

Contenido

3

Criptografía: Data Encryption Standard

M. en C. Víctor M. Silva G., M. en C. Eduardo Rodríguez Escobar, M en C. Eduardo Vega Alvarado
(CIDETEC-IPN)

Procesamiento de Imágenes: Estructura de Archivos BMP

Ing. Agustín Cruz Contreras, M. en C. Juan Carlos González Robles, M. en C. Juan Carlos Herrera Lozada
(CIDETEC-IPN)

6

9

Programadores para Microcontroladores PIC

M. en C. Juan Carlos González Robles, Ing. Agustín Cruz Contreras, M. en C. Juan Carlos Herrera Lozada,
(CIDETEC – IPN)

IC-Prog, Software para Prototipos de Programadores

M. en C. Juan Carlos González Robles, Ing. Agustín Cruz Contreras, M. en C. Juan Carlos Herrera Lozada,
(CIDETEC – IPN)

13

Sistema Autónomo para la Manipulación de Video en Tiempo Real Basado en un FPGA

Juan Diego Barranco, Enrique Guzmán Ramírez, Gabriel Gerónimo Castillo (Instituto de Electrónica y Computación,
Universidad Tecnológica de la Mixteca); M. en C. Romeo Urbieta Parrazales (CIC – IPN)

17

Diseño de Control Difuso Usando Promedio de Pesos e Implementado con Lenguaje Verilog

M. en C. Romeo Urbieta Parrazales (CIC-IPN), Enrique Guzmán Ramírez,
(Instituto de Electrónica y Computación, Universidad Tecnológica de la Mixteca)

23

Arquitectura Básica para Controladores de Lógica Difusa a Programarse en FPGAs

M. en C. Juan C. Herrera Lozada, Lic. Ma. de Lourdes Olvera Cárdenas, Lic. Ma. Teresa Lozano Hernández
(CIDETEC-IPN)

28

Tecnología VPN (1ª parte)

Lic. María Teresa Lozano Hernández, Lic. María de Lourdes Olvera Cárdenas, Ing. María del Rocío Velázquez Serrano
(CIDETEC-IPN)

33

39

Workflow: Tecnología Para la Innovación Organizacional

Lic. Elizabeth Acosta Gonzaga (CIDETEC-IPN), M. en C. Abraham Gordillo Mejia (UPIICSA-IPN)

PCI Express: Una Visión Inicial

M. en C. Eduardo Rodríguez Escobar, Lic. Elizabeth Acosta Gonzaga (CIDETEC-IPN)

44

Criptografía: Data Encryption Standard

*M. en C. Víctor M. Silva G.,
M. en C. Eduardo Rodríguez Escobar,
M. en C. Eduardo Vega Alvarado.
Profesores del CIDETEC-IPN.*

Desde la época del imperio romano, los hombres de estado deseaban enviar mensajes, ya sea al frente de batalla o a algún personaje importante, donde la condición era que el mensaje fuese secreto, esto es, que nadie aparte de ellos dos, el que envía y el que recibe, se enterara del contenido. En aquellos tiempos la encriptación y la desencriptación se trabajaba de forma manual y se hacía por medio de mascarillas, corrimiento de las letras del alfabeto, etc. En general se puede afirmar que hay tres tipos de elementos en todo proceso criptológico, llamémoslos A, B y C. El elemento A es el que desea enviar el mensaje, el elemento B el que lo recibe y C es el que desea conocer el contenido del mensaje que A envía a B.

En este trabajo se describe un criptosistema ampliamente utilizado en el mundo cuyo nombre es DES (Data Encryption Standard), siendo nuestro interés meramente educativo y con el fin de despertar en el alumnao la curiosidad por la Criptografía.

El criptosistema DES fue desarrollado por IBM y sale a la luz el 15 de enero de 1977, siendo revisado cada cinco años por el Buró Nacional de Normas de USA.

DESCRIPCIÓN DEL ALGORITMO

Se empezará con una descripción de alto nivel, consistente en presentar los aspectos importantes del criptosistema de una forma general y detallar mas adelante la mayoría de ellos. Por sencillez y para la mayor comprensión de DES, se considerará como texto claro a una cadena de 64 bits, tomando cuatro bits para cada carácter del sistema hexadecimal.

El algoritmo procede de acuerdo a los siguientes tres pasos:

Paso 1.- Se contará con una matriz PI de 8x8 (fija) la cual efectuará una permutación inicial sobre una cadena de texto claro X de 64 bits:

$$PI(X) = X_0.$$

La cadena X_0 se divide en dos subcadenas de 32 bits, pudiéndose representar como:

$$X_0 = L_0 R_0 ;$$

L_0 son los primeros 32 bits y R_0 son los 32 bits restantes.

Paso 2.- Se calcularán las subcadenas L_i, R_i en 16 iteraciones de acuerdo a la siguiente regla:

$$L_i = R_{i-1} \text{ y}$$

$$R_i = L_{i-1} \oplus f(R_{i-1}, K_i)$$

Aquí, \oplus representa la operación de "o exclusivo".

La representación gráfica de un ciclo de encriptación DES se muestra en la **figura 1**.

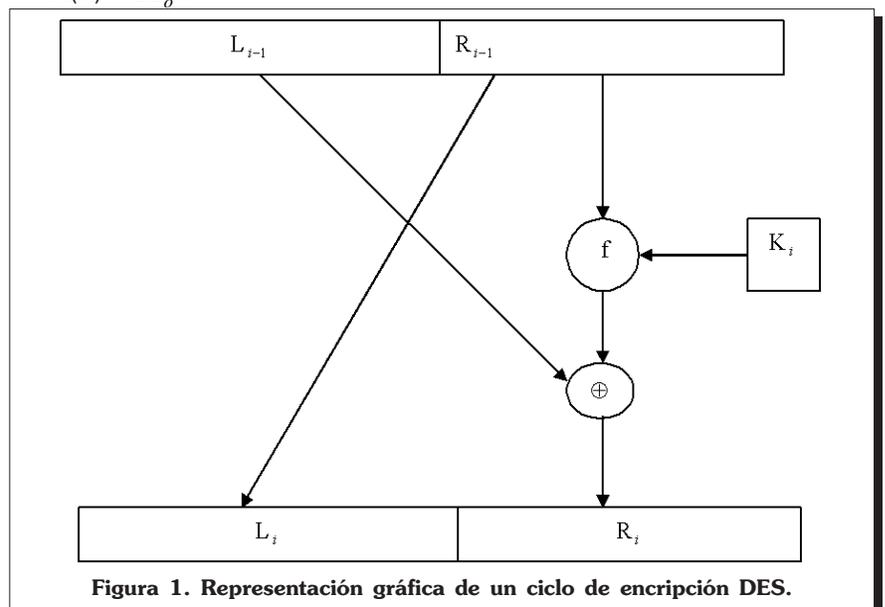


Figura 1. Representación gráfica de un ciclo de encriptación DES.

Paso 3.- Al resultado del ciclo 16 se invertirá el orden de aparición de las subcadenas $L_{16}R_{16}$; quedando la cadena $L_{16}R_{16}$, por último, se aplica la permutación inversa PI^{-1} a la cadena $R_{16}L_{16}$ para obtener finalmente el texto cifrado

$$Y=PI^{-1}(R_{16}, L_{16})$$

LA FUNCIÓN $f(R_{i-1}, K_i)$

Como se observó en el encabezado, la función f tiene dos argumentos R_{i-1}, K_i ; R_{i-1} es una cadena de 32 bits y K_i es una cadena de 48 bits. La función f produce a su vez una cadena de 32 bits. Sin que por el momento expliquemos como se obtiene la cadena de 48 bits de K_i , se describirán los cuatro pasos que se ejecutan para calcular el resultado de $f(R_{i-1}, K_i)$.

Paso 1.- El primer argumento de f, R_{i-1} , es expandido –permutado a una cadena de 48 bits. La función que expande y permuta $E(R_{i-1})$ es fija.

Paso 2.- Se efectúa la operación

$$E(R_{i-1}) \oplus K$$

y el resultado se puede pensar como una concatenación de 8 cadenas de 6 bits cada una. Entonces se puede escribir a

$$E(R_{i-1}) \oplus K \text{ como:}$$

$$B=B_1 B_2 B_3 B_4 B_5 B_6 B_7 B_8$$

Donde B_i es una cadena de 6 bits, con $1 \leq i \leq 8$

Paso 3.- Se contará con 8 cajas (matrices) de 4×16 ; S_1, S_2, \dots, S_8 . Los renglones de cada una de estas matrices será una permutación de los números enteros entre 0,15, además, estos arreglos cumplen con algunas otras propiedades que no mencionaremos en

este artículo. Con esta información se procederá de la siguiente manera:

Para cada B_i dada, el 1° y 6° bit definen el renglón de la matriz S_i y del 2° al 5° bit definen la columna de la matriz S_i . El resultado de esta operación es una sustitución, la cual se expresará como $C_i = S_i(B_i)$.

Como un ejemplo para ilustrar lo anteriormente mencionado, suponga que $i = 1$ con $B_1 = 101101$ y que la matriz S_1 está dada por:

$$S_1 = \begin{bmatrix} 14 & 4 & 13 & 1 & 2 & 15 & 11 & 8 & 3 & 10 & 6 & 12 & 5 & 9 & 0 & 7 \\ 0 & 15 & 7 & 4 & 14 & 2 & 13 & 1 & 10 & 6 & 12 & 11 & 9 & 5 & 3 & 8 \\ 4 & 1 & 14 & 8 & 13 & 6 & 2 & 11 & 15 & 12 & 9 & 7 & 3 & 10 & 5 & 0 \\ 15 & 12 & 8 & 2 & 4 & 9 & 1 & 7 & 5 & 11 & 3 & 14 & 10 & 0 & 6 & 13 \end{bmatrix}$$

Entonces $C_1 = S_1(101101) = 0001$

Paso 4.- La cadena $C = C_1 C_2 \dots C_8$ de longitud 32 bits es permutada por P (una matriz de permutación fija). El resultado $P(C)$ es definido como $f(R_{i-1}, K_i)$ para el ciclo i .

LAS LLAVES PROGRAMADAS K_1, K_2, \dots, K_{16}

En esta parte se verá como se obtienen las llaves K_1, K_2, \dots, K_{16} a partir de una llave K de 64 bits.

Paso 1.- Dada una llave K de 64 bits, se desechan los bits 8, 16, ..., 64 (8 en total), los cuales se conocen como bits de paridad, a los restantes 56 bits se les aplica una permutación $PC-1$ (fija), de hecho el orden del número de llaves es de 2^{56} . La cadena de 56 bits resultante se separa en dos subcadenas de 28 bits cada una, esto es,

$$PC-1(K) = C_0 D_0.$$

Paso 2.- Para $1 \leq i \leq 16$ se evalúan las siguientes expresiones :

$$C = LS_i(C_{i-1})$$

$$D_i = LS_i(D_{i-1})$$

$$\text{y } K_i = PC-2(C_i D_i)$$

LS_i representa un corrimiento a la izquierda de una o dos posiciones dependiendo del valor de i . Un caso particular sería correr una posición a la izquierda cuando $i = 1, 2, 9, 16$ y dos posiciones de otra manera. $PC-2$ es otra permutación fija.

Una mención importante:

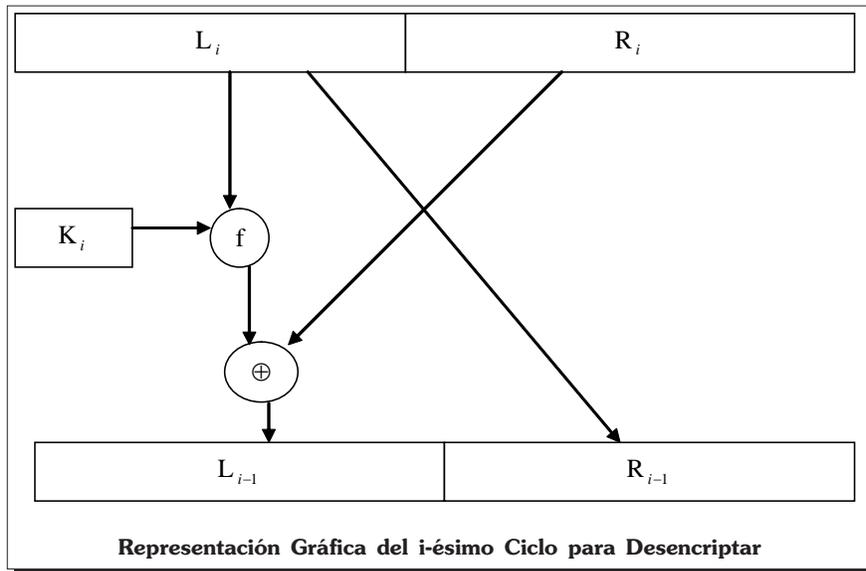
La permutación $PC-1$ se lleva a cabo en 56 bits y la permutación $PC-2$ desecha 8 bits y permuta los 48 restantes.

En este punto se ha descrito como procede DES para encriptar un texto claro, así como la forma de obtener las llaves programadas de 48 bits dada una llave de 64 bits; falta por describir como se descrypta un texto cifrado, lo cual se muestra en la **figura 2**.

A continuación se darán los resultados de encriptar varios textos claros, dos de ellos servirán para verificar que el programa que desarrollamos se ajusta a la norma internacional del criptosistema DES. Las cajas y permutaciones que se utilizaron están descritos en el trabajo [2].

Ejemplos:

TEXTO CLARO	LLAVE	TEXTO ENCRIPADO
0123456789ABCDE	133457799BBCDFF1	85E813540F0AB405
87878787878787	0E329232EA6D0D73	0000000000000000
SALVADOR	133457799BBCDFF1	8B2411C7EBCABAAF



Si el texto claro "SALVADOR" se desea expresar en código ASCII el resultado es el siguiente :

İ \$ Ä% Ä Û i% Q% «

CONCLUSIONES

Actualmente el criptosistema DES se considera poco seguro, claro está, si el valor de la información que se desea encriptar no pasa de determinada cantidad, se podría considerar a DES aún seguro, porque para encontrar la llave (2^{56} posibilidades) por el procedimiento exhaustivo, se necesitaría una supercomputadora con un valor de alrededor de un millón de dólares y la cual se llevaría un tiempo aproximado de 30 minutos. Una solución para resolver esta debilidad de DES es aplicar dos llaves como sigue: Se utiliza una primera llave K_1 para un texto claro dado y la salida de este proceso se toma nuevamente como texto claro para aplicar una llave K_2 . Por último, al resultado de este segundo proceso se le aplica la primera llave k_1 , el número de llaves posibles en esta situación es de 2^{112} .

BIBLIOGRAFÍA

- [1] Douglas R. Stinton, "CRIPTOGRAPHY: Theory and practice", CRC Press, 1995, USA.
- [2] Orlin Grabbe, "Data Encripyption Standard: The DES algorithm Illustrated", Laissez faire City Times Vol. 2, número 28, 2003. Página Web del autor: olingrabbe.org
- [3] Cevallos Fco. Javier, "C++: Enciclopedia del lenguaje", Alfaomega Ra-Ma, 2004, México.
- [4] Schildt Herbert, "LENGUAJE C: Programación Avanzada", Osborne/Mc Graw-Hill, 1987, México.

Procesamiento de Imágenes: Estructura de Archivos BMP

*Ing. Agustín Cruz Contreras,
M. en C. Juan Carlos González Robles,
M. en C. Juan Carlos Herrera Lozada
Profesores del CIDETEC-IPN.*

Lo presente trabajo describe la forma de interpretar la información contenida en archivos BMP, con el propósito de aplicar algoritmos propios de procesamiento a imágenes capturadas o generadas con sistemas comerciales.

INTRODUCCIÓN

Las imágenes se capturan o generan a través de equipos y programas comerciales, empleando formatos propios, como es el caso de Photoshop y Corel, o formatos con mayor difusión como BMP, JPG, PCX, etc.

En el desarrollo e implementación de algoritmos para el procesamiento de imágenes, se requiere la matriz de datos que contiene el color para cada pixel, esto no se tiene de manera directa si la imagen está almacenada con algún formato.

Una imagen puede almacenarse en un archivo siguiendo diferentes formatos, utilizando muchos de ellos compresión de datos. Cada uno tiene sus ventajas y desventajas, pero todos ellos tienen algunas características en común: Siempre se utiliza una cabecera en el archivo que identifica el tipo de formato del que se trata,

conteniendo información necesaria para interpretar el archivo (el tamaño de la imagen o el número de colores, etc.), después de la cabecera se encuentran los datos de la imagen, generalmente comprimidos con un algoritmo específico.

La imagen puede tener más o menos colores; entre más colores, será necesario un mayor número de bits por pixel para indicar el color del que se trata. Cuantos más colores, mejor calidad tendrá la imagen, pero de mayor tamaño será el archivo. Normalmente el número de colores es 16, 256 o 16 millones, lo que requiere 4, 16 o 24 bits por pixel. En el caso de utilizar 16 o 256 colores, debe especificarse a que color real corresponde cada uno de esos colores, es decir, que cantidades de Rojo, Verde y Azul serán utilizadas para representar el color en la pantalla. La tabla que asocia a cada color con las correspondientes cantidades de Rojo, Verde y Azul se llama paleta de colores. Puede ser modificada en función de la imagen, por lo que es necesario guardarla en el archivo. En el caso particular de utilizar 16 millones de colores, no se utiliza paleta de colores, pues la relación entre número de color y cantidad de Rojo, Verde y Azul es implícita: De los 24 bits por pixel, se utilizan 8 por color.

ALGUNOS FORMATOS DE IMAGEN

Con la intención de dar una idea clara sobre los formatos de imagen, a continuación se describen brevemente algunos de éstos.

BMP

El formato BMP (Windows BitMaP) es probablemente el más simple que existe, y consiste en una cabecera seguida por los valores de cada pixel, comenzando por la línea inferior y terminando por la superior, pixel a pixel de izquierda a derecha, manteniendo las líneas un tamaño múltiplo de 32. Su única ventaja es su sencillez. Su gran desventaja es el enorme tamaño de los archivos. Este formato se puede considerar como universal dado que prácticamente todas las aplicaciones lo pueden manejar.

PCX

En el formato PCX (de PC Paintbrush), los datos están comprimidos. Suponiendo que estemos utilizando 256 colores (un byte por pixel), el algoritmo consiste simplemente en reemplazar las secuencias de N pixels consecutivos del mismo color por dos bytes, de forma que el primero indique el número N de repeticiones y el segundo indique el color. Este algoritmo permite reducir el tamaño del archivo cuando la imagen sea un dibujo con planos de color constante, pues serán muchos los pixels consecutivos del mismo color. Sin embargo cuando haya algún pixel aislado de un color,

éste se reemplazará por los dos bytes. Por ese motivo es posible que en algunos casos llegue a aumentar el tamaño del archivo. Esto suele pasar con imágenes escaneadas, pues es improbable que varios pixels consecutivos tengan exactamente el mismo color.

GIF

El formato GIF (Graphic Interchange Format) fue inventado por Compu-serve. Utiliza un algoritmo de compresión similar al que usan los programas de compresión convencionales. Consiste en no detectar sólo las repeticiones de un color, sino en detectar las repeticiones de ciertas secuencias, mediante un diccionario que se va construyendo. Al igual que el formato PCX, funciona especialmente bien con imágenes en los que muchos pixels consecutivos tienen el mismo color, o se repiten secuencias de colores, pero también funciona bien con fotografías escaneadas o cualquier otra imagen. Además permite definir un color transparente, por lo que es muy útil para páginas Web.

JPG

El formato JPG utiliza un complejo algoritmo de compresión con pérdida de información. Es decir, el algoritmo modifica ligeramente los datos de forma que la imagen puede comprimirse mucho más. Gracias a esto el formato JPG es uno de los que más comprimen. Las modificaciones realizadas son inapreciables si se trata de una fotografía escaneada. Sin embargo, si se trata de un dibujo aparecen pixels visibles en la imagen. Por esta razón este formato es muy útil para fotografías escaneadas, pero no para dibujos, y por eso no permite definir colores como transparentes.

ESTRUCTURA DE UN ARCHIVO BMP

La estructura de un archivo BMP comprende las siguientes secciones:

- Encabezado
- Paleta de colores
- Mapa de bits

Con la intención de abordar el caso más simple se optó por manejar imágenes a 24 bits por color, con lo cual no se requiere de paleta de colores, y la información del color está dada por la combinación de rojo, verde y azul, en valores de 0 a 255.

La cabecera se integra por los siguientes campos:

<code>struct header{</code>	
<code> char type[2];</code>	2 bytes con los caracteres "BM"
<code> unsigned long size;</code>	4 bytes que indican el tamaño total del archivo
<code> char reserved[4];</code>	4 bytes reservados
<code> unsigned long offset;</code>	4 bytes que indican el offset desde el comienzo del archivo hasta el inicio del mapa de bits.
<code>};</code>	
<code>struct bmp_info{</code>	
<code> unsigned long bytes_in_header;</code>	4 bytes reservados
<code> unsigned long width;</code>	4 bytes que indican el ancho del gráfico (en pixels).
<code> unsigned long height;</code>	4 bytes que indican el alto del gráfico (en pixels)
<code> int planes;</code>	2 bytes que indican la cantidad de planos del gráfico
<code> int bits_per_pixel;</code>	2 bytes que indican la cantidad de bits por pixel
<code> unsigned long compression;</code>	Los siguientes campos de los cuales podemos prescindir en nuestra aplicación. formando un total de 54 bytes.
<code> unsigned long size_image;</code>	
<code> unsigned long h_resolution;</code>	
<code> unsigned long v_resolution;</code>	
<code> unsigned long n_indexes;</code>	
<code> unsigned long n_i_indexes;</code>	

Se debe tener presente que en el archivo los valores se organizan del byte menos significativo, al más significativo, y que la imagen se encuentra invertida, esto es: al inicio se encuentra la línea inferior, al final la línea superior, acomodadas de izquierda a derecha, con una longitud de línea en bytes múltiplo de 32. En caso de que la longitud de línea difiera de un múltiplo de 32, el programa de creación agrega los bytes faltantes para completar la longitud múltiplo de 32. En nuestro ejemplo la longitud de línea es de 10 pixels x 3=30 bytes, y el programa de creación agrega dos bytes con valor de cero para tener una longitud de 32 bytes.

PROGRAMA

El programa tiene como propósito extraer la información de la imagen, ubicando ésta en una matriz, que corresponde en dimensiones al alto y ancho de la imagen.

El programa se integra por los siguientes pasos:

- Apertura de archivo.
- Lectura de los campos "ancho y

alto": Con estos se determina el tamaño de la matriz que se requiere para almacenar la imagen.

- Lectura del campo "tamaño de cabecera": A partir de aquí comienza la información de la imagen.
- Lectura de datos considerando la organización descrita anteriormente.
- Despliegado de imagen en pantalla: Para verificar el éxito de la lectura.
- Cierre de archivo.

Ejemplo de estudio: Utilizando el "Paint" de Windows se formó una imagen de prueba con la composición mostrada en la **figura 1** (página siguiente).

Procesamiento de Imágenes: Estructura de Archivos BMP

R	R	R	R	R	R	R	R	R	R	G
V	V	V	V	V	V	V	V	V	V	G
A	A	A	A	A	A	A	A	A	A	G
N	N	N	N	N	N	N	N	N	N	G
B	B	B	B	B	B	B	B	B	B	G

Figura 1. Imagen sintética de prueba.

Donde:

R G B

R=Rojo FF-00-00
 V= Verde 00-FF-00
 A= Azul 00-00-FF
 N=Negro 00-00-00
 B=Blanco FF-FF-FF
 G= Gris 7F-7F-7F

42	4D	D6	00	00	00	00	00	00	00	00	36	00	00	00	28	00
00	00	0A	00	00	00	05	00	00	00	01	00	18	00	00	00	00
00	00	A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	FF										
FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
FF	7F	7F	7F	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	7F	7F	7F	00	00	FF	00									
00	00	FF	00	00	FF	00	00	FF	00	00	FF	00	00	FF	00	00
00	7F	7F	7F	00	00	00	00	FF	00	00	FF	00	00	FF	00	00
FF	00	00	FF	00	00	FF	00	00	FF	00	00	FF	00	00	FF	00
00	7F	7F	7F	00	00	00	00	FF	00	00	FF	00	00	FF	00	00
00	FF	00	00	FF	00	00	FF	00	00	FF	00	00	FF	00	00	00
FF	7F	7F	7F	00	00											

Figura 2. Volcado de archivo de imagen de prueba

Realizando el volcado del archivo en hexadecimal se tiene la información de la **figura 2**.

A cada casilla le corresponde un byte de información, contando las casillas, se tienen 214 bytes, que es el tamaño de archivo.

Asociando la información dada con los datos del archivo tenemos el esquema de mostrado en la figura 3.

Todas líneas mantienen la misma estructura, por lo que se considera suficiente, mostrar sólo el detalle de la primera en el archivo.

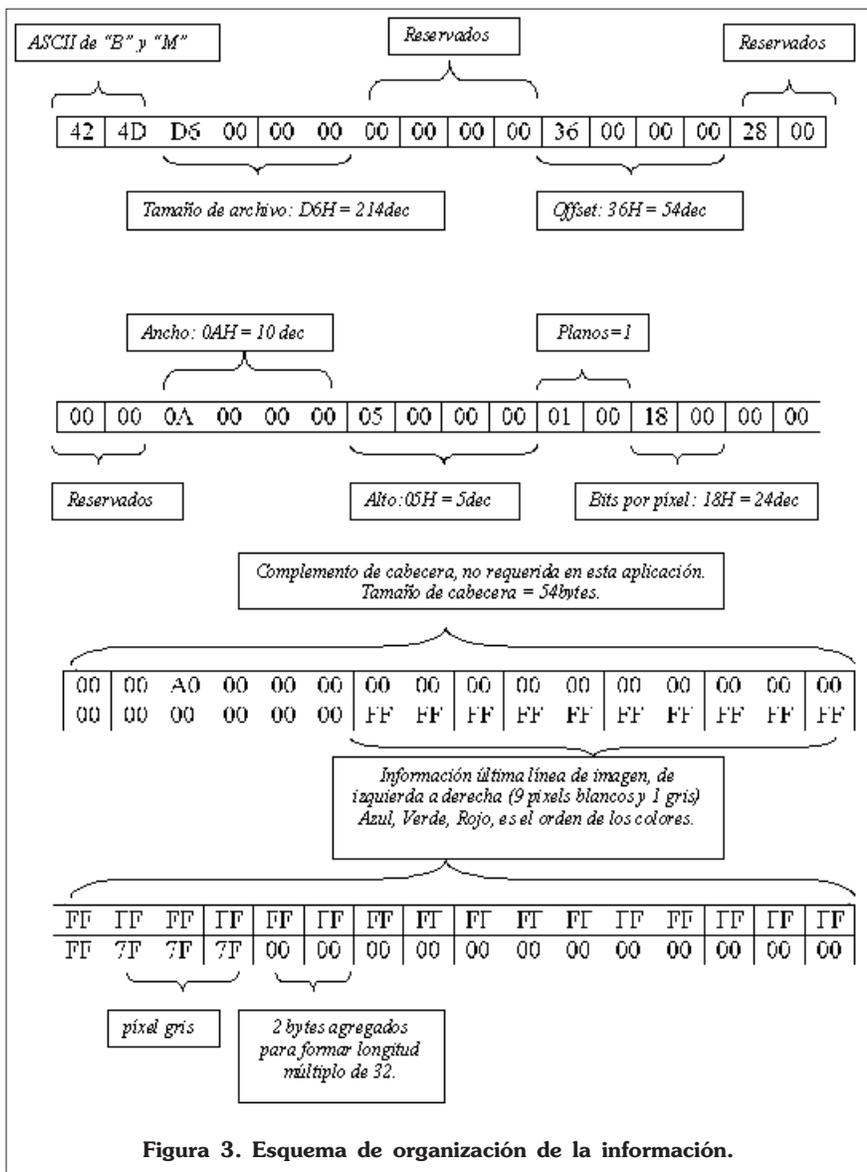


Figura 3. Esquema de organización de la información.

BIBLIOGRAFÍA

- [1] Steve Rimmer. "The Graphic File Toolkit". Addison Wesley. 1992.
- [2] Steve Rimmer. "Bit-Mapped Graphics". Mc Graw Hill. 1993.
- [3] David C. Kay, John R. Levine. "Graphics File Formats". Mc Graw Hill. 1995.

Programadores para Microcontroladores PIC

*M. en C. Juan Carlos González Robles,
Ing. Agustín Cruz Contreras,
M. en C. Juan Carlos Herrera Lozada,
Profesores del CIDETEC-IPN.*

Aunque la mayoría de los PICs (Peripheral Interface Circuits, Circuitos de Interfase Periférica) utilizan esencialmente el mismo sistema de programación, existen entre ellos sutiles diferencias, no sólo en cuanto a la disposición de las terminales (pines de conexión) utilizadas para la programación, sino también en cuanto a las señales eléctricas necesarias, lo que hace que algunos programadores sólo puedan programar ciertos tipos de PICs. Aquí se presentan dos programadores con distintas posibilidades y limitaciones, para que el lector elija el que se adapte mejor a sus necesidades.

ANTECEDENTES

Aun cuando no se pretende profundizar sobre los microcontroladores, es conveniente explicar qué son y cuáles son sus componentes. Un microcontrolador es un chip que, como mínimo, contiene en su interior un microprocesador, una memoria tipo ROM (que alberga el programa que ejecutará el microprocesador), una memoria RAM (para almacenar variables y datos), y unos puertos de entrada y/o salida que le permitirá

interactuar con otros dispositivos. Lo que diferencia a unos microcontroladores de otros son las características de cada uno de estos elementos y la inclusión o no de elementos adicionales, como memoria EEPROM, circuitería de reloj, temporizadores, contadores, convertidores AD, puertos serie, módulos PWM, etc. Cuando hablamos de programar un microcontrolador, básicamente nos referimos a grabar en la memoria de programa una serie de datos e instrucciones contenidos en un archivo, que corresponden al programa que ejecutará el microcontrolador durante su funcionamiento.

La utilización de microcontroladores en circuitos electrónicos comerciales se ha convertido en algo habitual desde hace unos años, debido principalmente a que permiten reducir el tamaño y el precio de los productos finales, pero hasta no hace mucho, resultaba poco menos que aterrador para el aficionado a la Electrónica incluirlos en sus montajes por diversas razones, como la dificultad para encontrarlos, su precio al por menor, la necesidad de complejos dispositivos para programarlos y lo arduo de escribir programas para ellos por la escasez y precio de herramientas software potentes. Además, la mayoría de los microcontroladores incorporaban una memoria de programa OTP (*One Time Programmable* – programable una sola vez), a la vez que los simuladores disponibles eran poco fiables, de forma que un

pequeño error en el código programado obligaba a desechar el microcontrolador.

En los últimos años, sin embargo, se ha abierto la vereda de los microcontroladores para los aficionados a la Electrónica, por varias razones. En primer lugar, el abaratamiento y el incremento de prestaciones han hecho que, aún siendo caros al por menor (un PIC16F84A cuesta industrialmente menos de 47.00 pesos, frente a los 80.00 que cuesta en los comercios), merezca la pena utilizarlos. Por otro lado, la política de varios fabricantes de ofrecer documentación y software de programación y simulación de forma gratuita, junto con cuestiones técnicas como la posibilidad de reprogramar los dispositivos (en muchos la memoria de programa es Flash, EPROM o EEPROM), y la simplicidad de los programadores necesarios (gracias a la implantación de protocolos de programación en serie frente a los antiguos sistemas de programación en paralelo), han hecho que hoy en día resulte muy fácil incluirlos en los diseños amateur.

Diversos fabricantes ofrecen gamas de microcontroladores para todos los gustos, aunque sin duda, el que se ha ganado el mercado para diseños sencillos (y buena parte de los profesionales) de los últimos tiempos es Arizona Microchip, con su conocida familia de microcontroladores PIC. Una de las razones es que todos utilizan el mismo juego de instruccio-

nes RISC (que además es muy reducido, son sólo 35 instrucciones distintas) y una filosofía de funcionamiento muy similares, de forma que si se aprende a utilizar un modelo concreto, es muy fácil adaptarse a otros, lo que resulta muy cómodo para el aficionado. Además, Microchip ofrece gratuitamente el editor y simulador de código **MPLAB**, bastante eficaz.

El objeto de este artículo es presentar una serie de programadores orientados especialmente a esta gama de microcontroladores. A lo largo del artículo se nombrarán diversos modelos de PICs. Para conocer a detalle sus características se recomienda visitar la página de Arizona Microchip, www.microchip.com.

DESARROLLO

Si bien son muchos los programadores disponibles en la Web que permiten programar microcontroladores PIC, se decidió tomar los más sencillos en cuanto a su implementación y elaboración (componentes económicos y fáciles de conseguir en nuestro país, software libre y fácil de usar y en español), y probarlos para así poder publicar el presente artículo o nota de aplicación de programadores fáciles y económicos para aficionados, estudiantes y docentes en el área de electrónica. La ventaja principal de hacerlo así es que son compatibles con gran parte del software de programación existente. Los dos programadores han sido probados con distintos PICs, lo que permite, en cada caso, enumerar sus posibilidades y limitaciones con bastante exactitud.

La razón de publicar los dos, es que el primero, si bien es limitado, su costo no pasa de 70.00 pesos y permite programar los modelos de PICs que se utilizan en la mayoría de los diseños prácticos, por lo que son idea-

les para quien necesite programar un PIC de forma constante, quedando el segundo, que tiene un costo que no pasa de 80.00 pesos, para los aficionados a la Electrónica que quieran hacer sus propios diseños y necesiten una herramienta flexible, segura y casi universal.

El primer programador es una versión modificada de uno de los muchos diseños de **JDM** (Diseñador Jens Dyekjaer Madsen). Tan sólo es capaz de programar los PICs 12C50x, 16x84, y las memorias EEPROM seriales 24Cxx. Utiliza el puerto serie para comunicarse con la PC, y de él obtiene los voltajes necesarios, por lo que no se necesita de una fuente de alimentación. La desventaja de este sistema es que si las señales del puerto serie no cumplen con la norma RS232, como ocurre en muchas portátiles, no funcionará correctamente. Sin embargo en la mayoría de las PCs de escritorio no habrá problema. Es compatible con el software IC-Prog (software libre para prototipos de programadores, el cual se encuentra en la dirección WEB: <http://www.ic-prog.com>), que permite realizar las operaciones necesarias (leer, borrar y grabar el PIC). Los detalles sobre este programador se encuentran descritos en el punto 3.1 Programador JDM2.

El segundo es una modificación del **Ludipipo** adaptada a la programación de diversos PICs. Al igual que el

anterior, para comunicarse con el PC utiliza el puerto serie, y de él toma los voltajes necesarios. Tiene, por lo tanto, el mismo problema que el diseño anterior al utilizarlo con PCs portátiles. Es compatible con el software IC-Prog y permite programar varios modelos de PICs, incluidos 16F8x, 16F62x, 16F87x, 18Fxx2 y 18Fxx8. Los detalles se encuentran descritos en el punto 3.2 Programador PIPO2.

PROGRAMADOR JDM2

EL CIRCUITO ESQUEMÁTICO

El programador que a continuación se presenta está dirigido a quien necesite programar alguno de los dispositivos mencionados con anterioridad. No es perfecto, falla en algunas PCs (portátiles sobre todo). Pero si todo lo que queremos es programar alguno de estos PICs, cuyo precio anda por los 80.00 pesos, tampoco estaría justificado gastar un dineral en un programador profesional (ya sea Microchip, Motorola, National Semiconductor, etc.) cuyo precio oscila de los 2,000.00 pesos en adelante. Este programador se puede construir sobre una placa fenólica universal (las placas perforadas para prototipos) de sólo 4 cm x 4.6 cm, en menos de una hora.

En la **figura 1**, se puede ver el circuito esquemático del programador. Sin profundizar en su funciona-

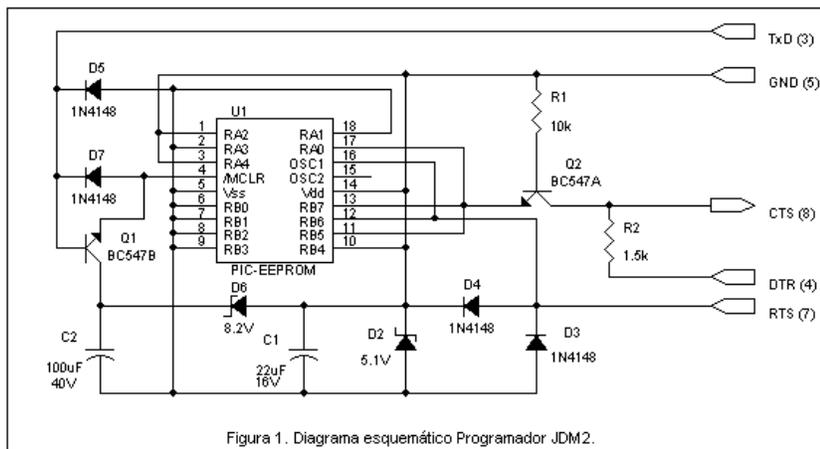


Figura 1. Diagrama esquemático Programador JDM2.

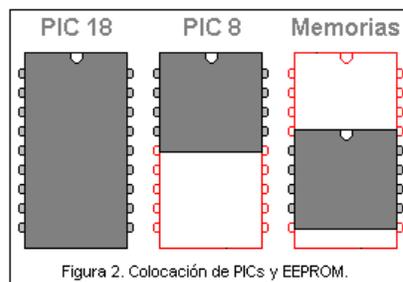
miento, para programar un PIC se necesitan dos voltajes respecto a tierra, +5V para la alimentación y +12.5V para la señal V_{pp} (voltaje de programación del PIC). La norma RS232 (puerto serie de la PC) establece que el voltaje correspondiente al 0 lógico debe estar entre -8V y -12V y el correspondiente al 1 lógico entre +8V y +12V. ¿Cómo conseguir entonces la señal V_{pp} necesaria para programar el PIC, que debe ser de +12.5V?

El truco está en crear una tierra para la alimentación del PIC que en realidad está a -5V respecto a la tierra del puerto serie. De esta forma, la tierra del puerto serie (pata 5 del conector DB9) queda a +5V respecto a la tierra del PIC (pata 5 o 18 de la base para cualquiera de los PICs), y de hecho, es la tierra del puerto serie la que se utiliza como alimentación de +5V para el PIC. El condensador C1 se carga mediante los diodos D3 y D4, y sirve de fuente de voltaje de +5V, que queda estabilizada mediante el diodo zener D2. De forma parecida, C2 se carga mediante D5 y Q1, quedando su voltaje estabilizado a unos +13V gracias a la unión en serie de los diodos zener D6 y D2. Este voltaje es el que se utiliza como voltaje de programación para el PIC.

Para las demás señales necesarias para la programación, DATA y CLOCK se utilizan directamente las señales del puerto serie CTS (DATA desde el PIC), DTR (DATA hacia el PIC) y RTS (CLOCK hacia el PIC).

Esta filosofía es la utilizada en la mayoría de programadores que toman su alimentación del puerto serie, pero la mayoría fallan a la hora de programar los PIC por las especiales características de este PIC en cuanto a temporización y consumo de V_{pp} . Este diseño, se probó con el software libre de desarrollo IC-Prog versiones 1.04c y 1.05a.

La colocación de los diferentes dispositivos a programar se muestra en la figura 2. Se dispone de un zócalo de 18 pines el cual permite conectar, tanto PICs como memorias EEPROM seriales. Los PIC's de 18 pines sólo se colocan ocupando la totalidad del zócalo mientras que los PIC's de 8 pines se colocan ocupando la parte superior del mismo. En tanto las memorias seriales se colocan una hilera anterior al fin del zócalo (el pin 1 de la memoria debe coincidir con el 5 del zócalo).



PROGRAMADOR PIPO2 (LUDIPIPO)

EL CIRCUITO ESQUEMÁTICO

Este programador, basado en el Ludipipo, está adaptado a la programación de PICs, es simple y barato, pero permite programar a través del puerto serie de cualquier PC de escritorio (en las portátiles da problemas) una cantidad nada despreciable de PICs distintos, y todo ello utilizando software libre de programación estándar, como IC-Prog 1.05A.

Se pueden programar sin problemas los siguientes PICs:

16F627, 16F628.
16C84, 16F83, 16F84.
16F873, 16F874, 16F876, 16F877.
18F242, 18F252, 18F258, 18F442,
18F452, 18F458.

Y, al menos en teoría (no se han probado) debería funcionar correctamente con los siguientes:

16C62, 16C63, 16C64, 16C65,
16C66, 16C67.

16C71, 16C72, 16C73, 16C74,
16C75, 16C76, 16C77, 16C715.
16C620, 16C621, 16C622,
16C623, 16C624, 16C625.
16F870, 16F871, 16F872.
16C923, 16C924.

Sin entrar en detalles sobre el funcionamiento del circuito esquemático de la **figura 3**, diremos sólo que se basa en principios muy parecidos a los del JDM, pero con ciertos retoques en la temporización y la estabilidad de las señales. En este caso, se utiliza como alimentación de +5V el condensador C2, que se carga mediante D2, D3 y D4 en los momentos en que las señales DTR, RTS y TxD del puerto serie son negativas, y su voltaje queda estabilizado mediante D7. D5 fija la tensión de programación en +12V y D6 limita la tensión de la señal CLOCK a +5V.

La colocación de los diferentes dispositivos a programar se muestra en la **figura 4**. Se dispone de un zócalo compuesto de tres tiras de pines el cual permite conectar, tanto PIC's de 40, 28 y 18 pines. Los PIC's de 40 pines se colocan ocupando la totalidad del zócalo (quedando libre la tira de en medio debajo del PIC) mientras que los PIC's de 28 pines se colocan ocupando la parte superior del zócalo (quedando el pin 1 sobre la tira de en medio, y el pin 28 en lo que sería el pin 40 del de 40), en tanto que los de 18 pines quedan montados también sobre la tira de en medio, pero sobre la parte inferior quedando el pin 9 en el lugar 20 de esta tira, y el pin 10 en lo que sería el pin 21 del PIC de 40.

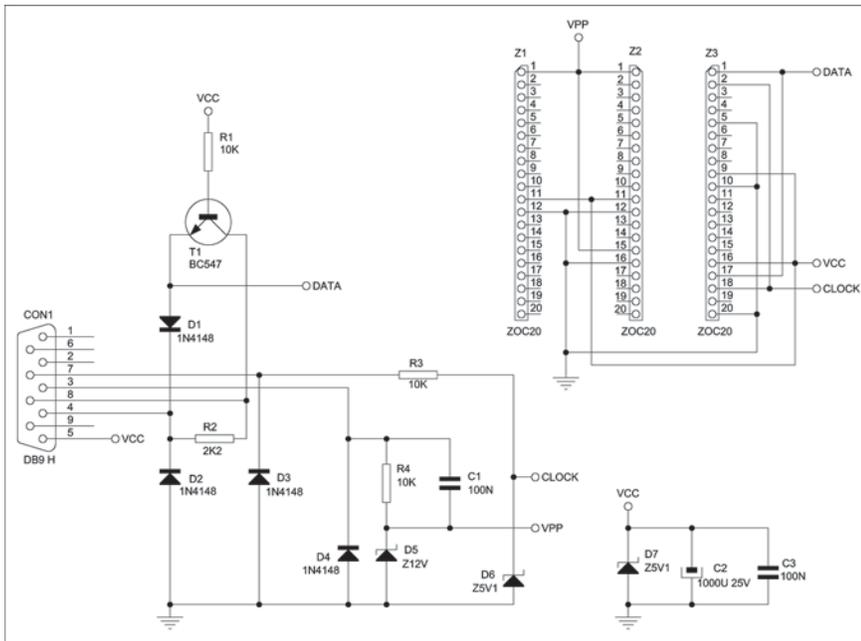


Figura 3. Diagrama esquemático del Programador PIPO2.

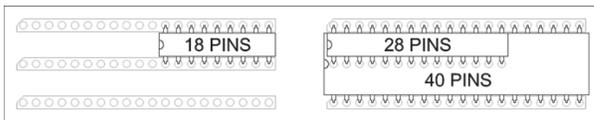


Figura 4. Colocación de PICs en sus diferentes tamaños.

CONECTOR

Para conectar el programador a la PC hará falta también un cable serie transparente (cableado pin a pin) que tenga cableados al menos los pines que se indican en la **figura 5**. Este cable se puede fabricar o comprarlo ya hecho (es el tipo de cable que se utiliza para conectar un modem a la PC).

CONCLUSIONES

Lo primero que hay que tener claro a la hora de utilizar estos programadores es el orden en que se deben hacer las cosas para no estropear ni el programador, ni el PIC, ni el puerto serie de la PC. Siempre que queramos insertar o extraer un PIC del zócalo hay que desconectar el programa-

del puerto serie, ya que, al extraer la alimentación del puerto serie, mientras esté conectado estará alimentado. Por lo tanto, el proceso a seguir consta de los siguientes pasos:

- 1º. Con el programador desconectado insertar el PIC en el zócalo en la posición correcta (figuras 2 o 4).
- 2º. Conectar el programador al cable que viene del puerto serie de la PC.
- 3º. Llevar a cabo las operaciones de grabación o lectura necesarias.
- 4º. Desconectar el programador del cable que viene del puerto serie de la PC.
- 5º. Extraer el PIC del zócalo.

Por otro lado, es necesario instalar y configurar adecuadamente el software para poder interactuar con el programador (JDM2 o PIPO2). Esta instalación se detalla en el artículo "IC-Prog software para prototipos de Programadores".

BIBLIOGRAFÍA E INTERNET

- [1] José M^a. Angulo Usategui, Ignacio Angulo Martínez. "Microcontroladores PIC: Diseño práctico de aplicaciones". Editorial Mc Graw Hill, 2^a. Edición, 1999.
- [2] José M^a. Angulo Usategui, Susana Romero Yesa, Ignacio Angulo Martínez. "Microcontroladores PIC: Diseño práctico de aplicaciones. Segunda Parte: PIC16F87X". Editorial Mc Graw Hill, 2^a. Edición, 2000.
- [i] www.microchip.com
- [ii] www.ic-prog.com
- [iii] www.jdm.homepage.dk/newpic.htm
- [iv] club.telepolis.com/vayas10/programadores.htm
- [v] 213.97.130.124/progs/progs.htm
- [vi] www.redeya.com
- [vii] www.geocities.com/pictarjetas/esquema3.htm
- [viii] www.msebilbao.com/tienda/default.php
- [ix] skyscraper.fortunecity.com/email/250/pic.htm
- [x] www.pic-compiler.com/hw/p_piccolog.htm
- [xi] www.blichfeldt.dk/propic/

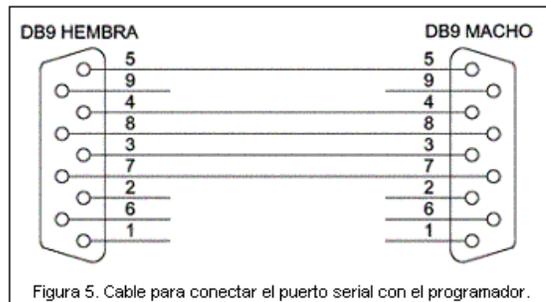


Figura 5. Cable para conectar el puerto serial con el programador.

IC-Prog, Software para Prototipos de Programadores

*M. en C. Juan Carlos González Robles, Ing. Agustín Cruz Contreras,
M. en C. Juan Carlos Herrera Lozada,
Profesores del CIDETEC-IPN.*

El IC-Prog es un software de programación basado en ventanas bajo ambiente Windows que permite la programación de diversos dispositivos y soporta numerosos tipos de programadores. El IC-Prog maneja archivos en formato Intel Hexadecimal (*.hex) de 8 y 16 bits, archivos objeto (*.obj) y archivos binarios (*.bin).

Tipos de archivo soportados actualmente:

- IHX8 (*.hex, *.h8, *.sxh)
- IHX16 (*.hex, *.h16, *.sxh)
- Binario (*.bin)
- Objeto (*.obj)
- Motorola S (*.mot, *.s19)
- Archivos eeprom (*.e2p, *.eep)
- Cualquier archivo (*.*)

Para que el programa funcione se deberá conectar a la computadora un programador, y configurar correctamente tanto a éste como al programa. Debido a la gran variedad de programadores y sus diferencias, el programa puede no funcionar con ciertos Sistemas Operativos y equipos programadores.

El IC-Prog requiere Windows 95, 98, ME, NT, 2000 o XP y un coprocesador interno o externo para funcionar. Todos los procesadores compatibles y superiores a un 386 con 8Mb de memoria RAM. Este es un programa registrado con libre distribución. Se permite su copia y distribución en la medida en que no se lo modifique de manera alguna, se cobre por su uso o se utilice para fines ilegales.

El IC-Prog ha sido designado como una aplicación de programación universal para todos los programadores. Si se posee un diseño propio de un programador y se desea que el mismo sea soportado por el IC-Prog, es necesario contactar al autor.

INSTALACIÓN

La instalación de este software es muy sencilla, y basta con descomprimir el archivo ICProg.zip, que se puede encontrar en la página web www.ic-prog.com, este archivo consta del archivo Icprog.exe que contiene todo el código necesario para su funcionamiento, con versiones de Windows 95, 98, ó Millenium. También puede obtenerse el archivo de ayuda Icprog.hlp (en español), si bien este archivo no es necesario para el funcionamiento del programa, si es de ayuda para saber cómo usar el menú del programa.

I. Para que este programa funcione correctamente en Windows XP Home Edition, ó XP Profesional es necesario descargar de esta misma página IC-Prog NT/2000 driver e instalar el archivo Icprog.sys para compatibilidad con Windows 2000, NT y XP y debe ser puesto en la misma carpeta que el Icprog.exe y el Icprog.hlp, como se muestra a continuación.



II. Una vez instalados los tres archivos en la misma carpeta (por ejemplo C:\Icprog\Icprog.exe), al ejecutar este archivo por primera vez nos aparecerá una ventana como se muestra en la **figura 1**. En esta ventana, seleccionamos OK y continuamos.

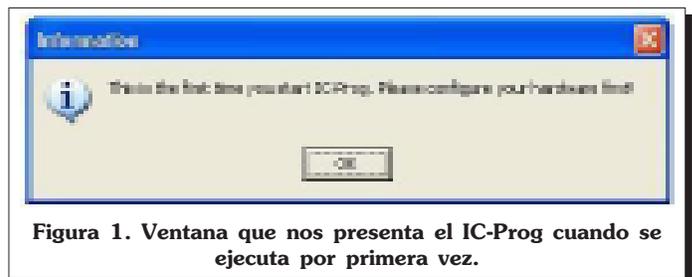


Figura 1. Ventana que nos presenta el IC-Prog cuando se ejecuta por primera vez.

III. Al seleccionar OK, se nos solicitará configurar los parámetros del hardware (Hardware Settings) para su correcto funcionamiento (dependerá del tipo de programador que tengamos y el puerto en que esté conectado, para nuestro caso es JDM y COM1 respectivamente), así como el tipo de interfase, que será Direct I/O, y por último le pondremos un retardo de I/O de 10 (puede variar dependiendo del tipo de procesador que tengamos en nuestra PC), y seleccionamos OK, como se muestra en la **figura 2**.



Figura 2. Ajustes de Hardware.

IV. Al realizar lo anterior, se obtiene un mensaje de instrucción privilegiada y seleccionamos OK, como se muestra en la **figura 3**.



Figura 3. Mensaje posterior a la configuración.

V. Después de esto, nos presenta la ventana principal del programador de prototipos con un mensaje de Violación de acceso en la dirección 0046XXXX del módulo icprog.exe, aquí pulsamos "Aceptar" para poder entrar al software de programación de PICs, como se muestra en la **figura 4**.

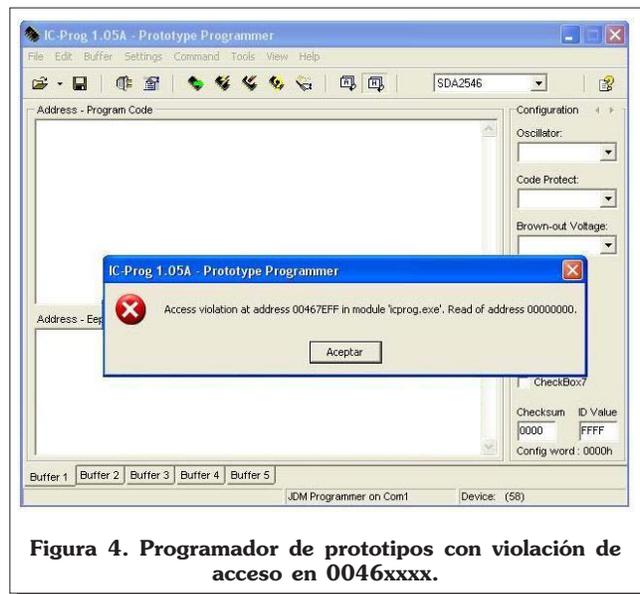


Figura 4. Programador de prototipos con violación de acceso en 0046xxxx.

VI. Una vez dentro del software, nos salimos de él seleccionando en el menú File y posteriormente Exit, o bien lo cerramos seleccionando el tache del extremo superior derecho, como se muestra en la **figura 5**.

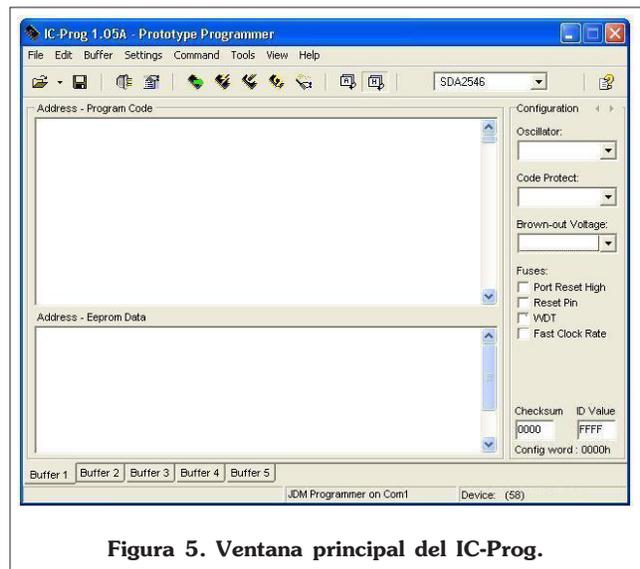


Figura 5. Ventana principal del IC-Prog.

VII. El paso siguiente es configurar las propiedades del Icprog para que sea compatible con Windows 2000, XP y NT, esto lo realizamos accediendo al archivo donde se encuentra el Icprog.exe, y con el botón derecho del mouse seleccionamos propiedades y en la pestaña Compatibilidad seleccionamos Windows 2000, y seleccionamos "Aceptar", ver **figura 6**.

VIII. Una vez que se hace compatible al software, se vuelve a ejecutar el Icprog, que presenta de nuevo una



Figura 6. Compatibilidad con Windows 2000.

ventana como la de la **figura 3**, en la cual se selecciona OK.

IX. La ventana que aparece enseguida es como la que se muestra en la **figura 7**, donde lo primero que hay que hacer es cargar el driver del Icprog seleccionando en el menú "Settings" y en el submenú "Options" o en el icono  "Options" de acceso rápido la pestaña "Misc", donde habilitamos la casilla "Enable NT/2000/XP Driver", con lo que abre otra ventana que notifica que se tiene que reiniciar el Icprog para que el driver trabaje, por lo cual se selecciona "Yes" para reinicializarlo.

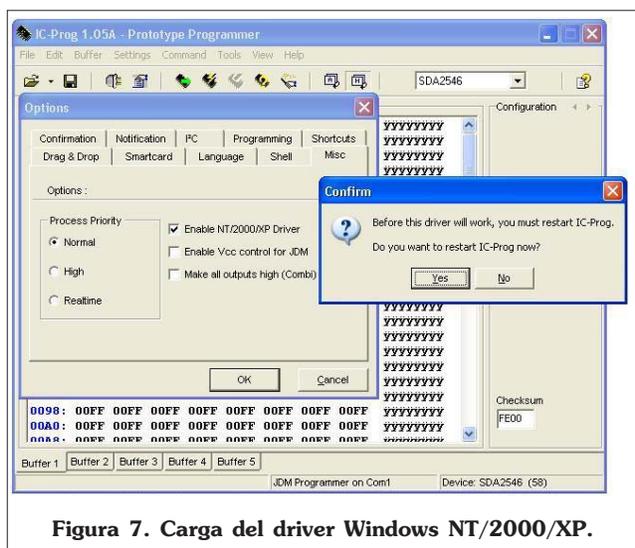


Figura 7. Carga del driver Windows NT/2000/XP.

X. Al reiniciar el IC-Prog, se presenta una ventana que notifica que no está instalado el driver en la PC, y que si se desea instalar, a lo cual seleccionamos "Yes". Una vez instalado, se abre la ventana del programador en la cual ya podemos modificar los parámetros que deseamos de acuerdo a nuestras necesidades, una por ejemplo puede ser cambiar el idioma al español, la cual se hace seleccionando el icono "Options" de acceso rápido, y enseguida la pestaña de "Language" en la cual seleccionamos "Spanish" y posteriormente "OK".

XI. Con los pasos realizados hasta el momento, ya se ha instalado el software del programador IC-Prog para poder programar cualquiera de los microcontroladores PIC que soporte el hardware que tengamos, para esto sólo tenemos que seleccionar en el menú "Ajustes" y en el submenú "Tipo hardware F3" ó seleccionamos el icono de acceso rápido  ó tecleamos "F3" para configurar el tipo de programador, el puerto, la interfaz y el retardo, como se puede ver en la figura 3.

XII. Por último, para poder programar un microcontrolador en específico, se tendrá que seleccionar el tipo de PIC, así como se tendrá que cargar el archivo *.hex (que es el archivo *.asm de nuestro diseño ya ensamblado como por ejemplo en el MPLAB), y además se tendrán que configurar todos los bits del dispositivo correspondiente, ver **figura 8**. Para una explicación más detallada sobre el uso del IC-Prog, consultar el archivo de ayuda icprog.chm.

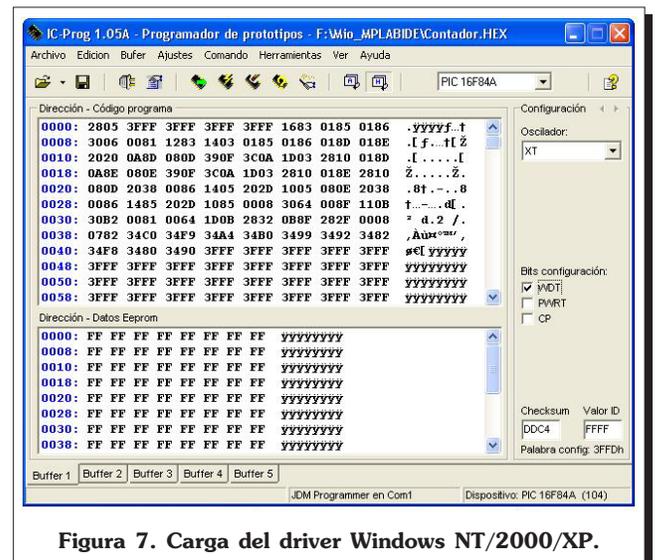


Figura 7. Carga del driver Windows NT/2000/XP.

PÁGINAS DE INTERNET CONSULTADAS

- i.** <http://www.microchip.com>
- ii.** <http://www.ic-prog.com>
- iii.** [http://autric.com/Microbotica y Mecatronica/herramientas.htm](http://autric.com/Microbotica_y_Mecatronica/herramientas.htm)
- iv.** http://kudelsko.free.fr/prog_pic_rs232/PIC_XP.htm
- v.** <http://www.jdm.homepage.dk/newpic.htm>
- vi.** <http://club.telepolis.com/vayas10/programadores.htm>
- vii.** <http://213.97.130.124/progs/progs.htm>
- viii.** <http://www.redeya.com>
- ix.** <http://www.geocities.com/pictarjetas/esquema3.htm>
- x.** <http://www.msebilbao.com/tienda/default.php>
- xi.** <http://skyscraper.fortunecity.com/email/250/pic.htm>
- xii.** http://www.pic-compiler.com/hw/p_piccollog.htm
- xiii.** <http://www.blichfeldt.dk/propic/>

Sistema Autónomo para la Manipulación de Video en Tiempo Real Basado en un FPGA

*Juan Diego Barranco, Enrique Guzmán Ramírez, Gabriel Gerónimo Castillo,
Instituto de Electrónica y Computación
Universidad Tecnológica de la Mixteca
e-mail: jeanknoj1@yahoo.com*

*Romeo Urbieto Parrazales
Profesor del CIC-IPN
e-mail pollux.cic.ipn.mx*

El presente trabajo tiene como objetivo el diseño e implementación de una plataforma hardware autónoma para la manipulación de video en tiempo real basada en una matriz de compuertas programable en el campo, el enfoque del sistema está dirigido a aplicaciones de tratamiento digital de imágenes. Para este fin se propone el uso de la tecnología de los dispositivos lógicos programables como parte central del sistema, planteando la posibilidad de realizar un sistema de captura y procesamiento de imágenes a alta velocidad, con base en la gran flexibilidad y la velocidad de un hardware dedicado. La implementación de los algoritmos de control, manipulación e interpretación de la información de video se hace en un FPGA de la firma Xilinx mediante el lenguaje descriptor de hardware VHDL.

INTRODUCCIÓN

Los digitalizadores de video, también conocidos como "frame grabbers", se fabrican en muchas formas y tamaños con cantidades variantes de funciones incluyendo diferentes conexiones a la computadora, ya que puede ser mediante el bus ISA (**I**ndustry **S**tandard **A**rchitecture), PCI (**P**eripheral **C**omponent **I**nterconnect), USB (Universal Serial Bus), PS/2, Serial, Paralelo o un conector RCA, entre otros [1]. Este tipo de sistemas hace necesario el uso de una computadora personal (PC), la cual alberga los algoritmos de adquisición y manipulación de la información de video, obteniendo sistemas costosos e inmóviles.

Por otra parte, el uso de una herramienta para la manipulación de video cada vez es más común en aplicaciones autónomas, tales como robótica, medicina y seguridad, entre otras [2]. Este tipo de sistemas autónomos basan su funcionamiento en microcontroladores y procesadores digitales de señales (DSP, Digital Signal Processor)[URL1].

Una alternativa más para este tipo de sistemas es el uso de dispositivos lógicos programables (PLD, Programmable Logic Device) debido a su gran flexibilidad y altas velocidades de ejecución, así como a la gran variedad de herramientas de desarrollo que responden a una filosofía de diseño de sistemas digitales complejos[3].

La herramienta de diseño elegida para el desarrollo de este sistema fue Xilinx Foundation en su versión 4.2i [URL2], el sistema fue implementado en la tarjeta XS40 de la firma Xess Corp. [URL3], la cual cuenta con un FPGA XC4010XL de Xilinx [4], la implementación del sistema tiene la capacidad de adaptarse a cualquier FPGA.

La descripción de los diferentes módulos que componen al sistema fue hecha utilizando el lenguaje descriptor de hardware VHDL (VHSIC Hardware Description Language), debido a su versatilidad, portabilidad y ser un lenguaje estandarizado [5,6]; VHDL se encuentra completamente especificado en los estándares IEEE 1076-1987 y 1164 [7].

El objetivo de este trabajo es desarrollar un sistema autónomo de adquisición de imágenes de bajo costo que permita la aplicación de algoritmos de procesamiento y análisis de imágenes.

ESQUEMA GENERAL DEL SISTEMA AUTÓNOMO PARA LA MANIPULACIÓN DE VIDEO

Considerando las características más comunes que componen un digitalizador de video y las necesidades que

se requieren para las aplicaciones que se dará a este trabajo, se obtiene un esquema general mostrado en la **figura 1**, el cual consta de los siguientes módulos:

- **Circuiteria analógica.** Tiene como funciones el acoplamiento de video al sistema, corrección en CD de la señal de video, rechazo a la subportadora de color, almacenamiento parcial de la señal, generación de voltajes de referencia para el convertidor analógico digital.
- **Separador de sincronía.** Basa su funcionamiento en el circuito integrado LM1881, extrae información de coordinación de la señal de video, incluyendo sincronización compuesta y vertical, coordinación de pórtico anterior e información de los campos par e impar desde una fuente de video NTSC, PAL o SECAM.
- **Memoria de almacenamiento.** Es un bloque de memoria de 1024 Kbytes, sirve como espacio de almacenamiento de la información de video adquirida para su posterior manipulación.
- **Digitalización.** Consta del convertido analógico a digital ADS802U de 12 bits, el cual tiene una velocidad de muestreo de 10 millones de muestras por segundo.
- **Interfaz con una PC.** Aunque se ha mencionado que se trata de un sistema autónomo, fue necesario implementar una interfaz con una PC para poder visualizar los resultados de la adquisición y manipulación del video. La interfaz con la PC se realizó por el puerto serie, en el sistema fue necesario implementar el protocolo RS232 dentro del FPGA.
- **Etapas de control y manipulación del video.** Es la parte central del sistema y basa su funcionamiento en un FPGA XC4010XL de la firma Xilinx. Consta de 3 módulos principales implementados en VHDL: protocolo RS232, etapa de control e implementación de algoritmos de procesamiento digital de señales.

Las especificaciones del sistema autónomo para la manipulación de video son las siguientes:

- Capacidad para adquirir marcos o campos de cualquier fuente de video reglamentada por el estándar RS-170.
- Soporta el formato NTSC.
- Digitalizar imágenes de hasta 488 líneas horizontales por 785 líneas verticales.
- Soporta hasta 4096 niveles de grises.
- Almacenar imágenes dependiendo de su resolución, para su posterior transmisión hacia la PC o para procesarla mediante un algoritmo de procesamiento de imágenes.

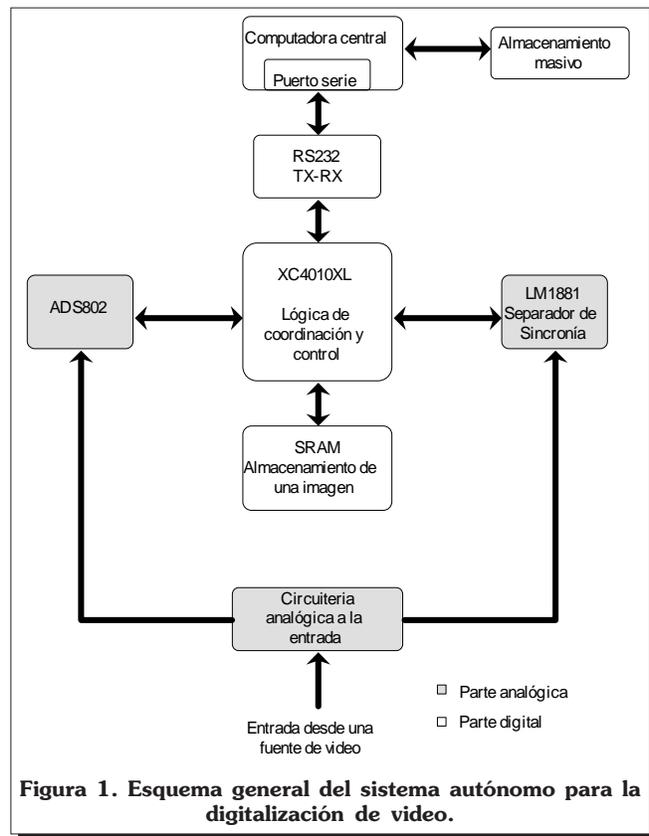


Figura 1. Esquema general del sistema autónomo para la digitalización de video.

- Digitaliza imágenes monocromáticas, cuando sean utilizadas fuentes de video a color las imágenes resultantes serán monocromáticas ya que la información de color se elimina de la señal de video compuesto antes del proceso de digitalización.
- Permite enviar imágenes completas a una PC por medio de una conexión RS232 para evaluar el funcionamiento del digitalizador y no para realizar procesamiento en la PC.
- El FPGA utilizado implementa algoritmos de procesamiento de imágenes.

Todos y cada uno de estos módulos están contenidos en las etapas de Acondicionamiento de la señal de video y, de Control y manipulación de video.

ACONDICIONAMIENTO DE LA SEÑAL DE VIDEO

Esta parte del sistema es analógica, se trata de una versión modificada de la que originalmente fue presentada por Craig A. Lindley [8]. Ésta corresponde en parte al tratamiento de la señal de video y a los dispositivos empleados. Las funciones realizadas en esta etapa incluyen:

- Acoplamiento de video con terminaciones de 75W.
- Corrección en CD de la señal de video.
- Rechazo a la subportadora de color.
- Almacenamiento parcial de la señal
- Generación de los voltajes de referencia para el ADC.

La parte central de esta etapa esta compuesta por el separador de sincronía de la señal de video LM1881. El circuito implementado con este circuito se ilustra en la **figura 2**. Este circuito integrado extrae información de coordinación de la señal de video, incluyendo sincronización compuesta y vertical, coordinación de pórtilo anterior e información de los campos par e impar desde una fuente de video NTSC, PAL o SECAM con amplitudes desde 0.5Vpp hasta 2.0Vpp. La salida vertical se produce en el rizo de subida del primer pulso en el periodo de sincronía vertical.

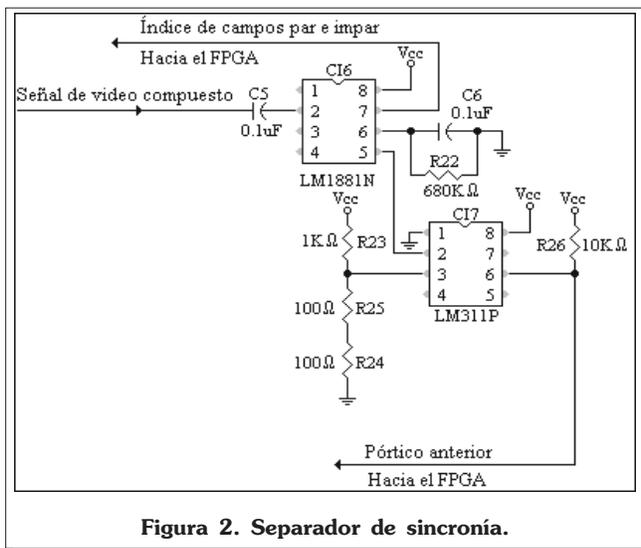


Figura 2. Separador de sincronía.

Aunque el LM1881 proporciona cuatro señales de sincronía especializadas, este diseño solo utiliza la salida de campos par e impar y el pulso de pórtilo anterior. Este par de salidas generan señales de coordinación que identifican los campos de video para el almacenamiento en memoria, recuperan señales de sincronización contaminadas u omitidas y también proporcionan referencias de coordinación para la extracción de datos codificados o no codificados en líneas de video específicas. Para entender mejor la información de coordinación y los tipos de señales del LM1881, éstas se ilustran en la **figura 3**.

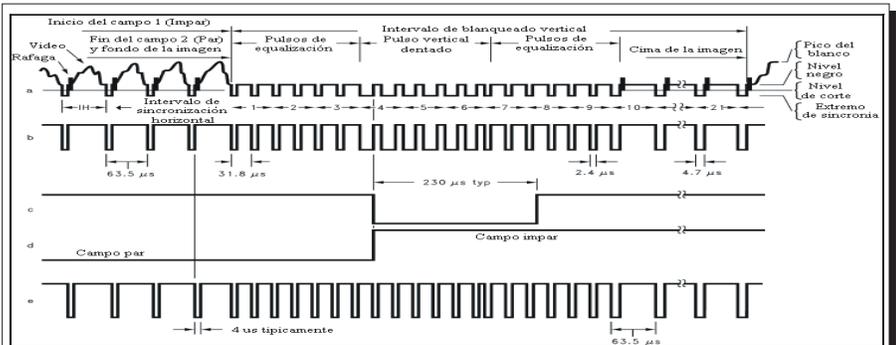


Figura 3. Señales controladas por el LM1881. (a) Video compuesto. (b) Sincronización compuesta. (c) Pulso de salida vertical. (d) Índice de campos par e impar. (e) Señal de pórtilo anterior.

CONTROL Y MANIPULACIÓN DE VIDEO

Esta etapa está implementada en un FPGA XC4010XL de la firma Xilinx, utilizando el lenguaje descriptor de hardware VHDL, y está dividida en tres módulos: módulo de control y sincronía, módulo protocolo RS232 y módulo de manipulación de video (**figura 4**).

MÓDULO DE CONTROL Y SINCRONÍA

Este módulo implementa el protocolo mediante el cual el FPGA XC4010XL controla la adquisición de la imagen de una fuente de video NTSC a través del convertidor analógico a digital ADS802U de la compañía Texas Instruments. El ADS802U es un ADC monolítico de 12 bits, éste opera a baja potencia y en un rango de muestreo máximo de 10 millones de muestras por segundo. Algunas de sus ventajas son que opera con una sola fuente de alimentación de 5V y se puede configurar para aceptar señales diferenciales o referenciadas a tierra. Emplea corrección de errores vía digital para proveer una excelente ejecución lineal, baja distorsión, alta relación señal a



Figura 4. Distribución de recursos del FPGA XC4010XL.

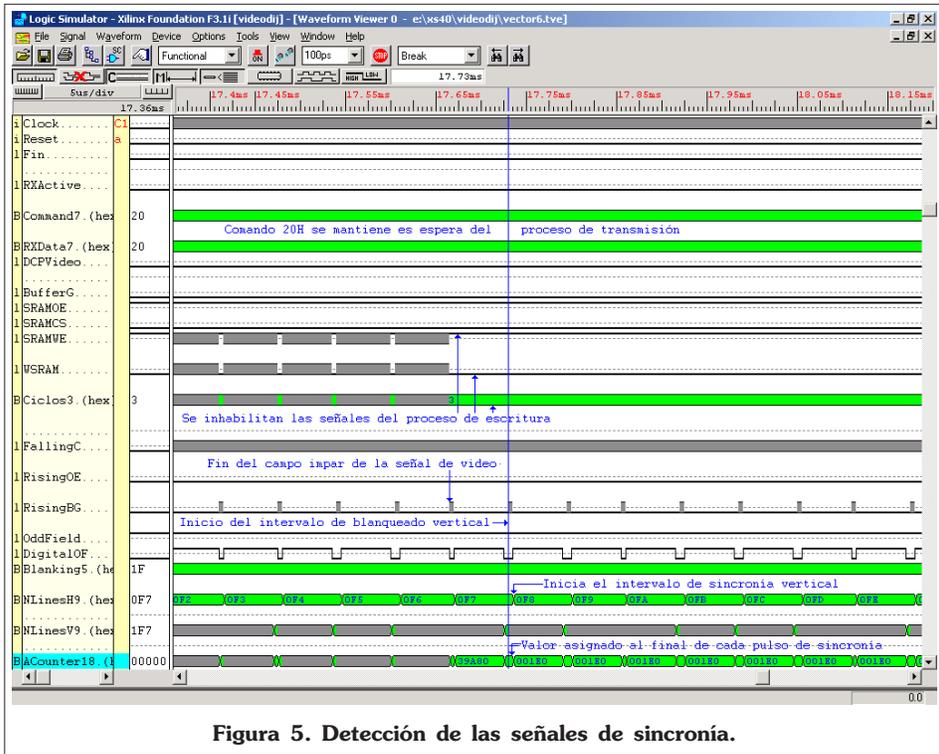


Figura 5. Detección de las señales de sincronía.

MÓDULO PROTOCOLO RS232

Al tratarse de un sistema autónomo carece de un sistema de despliegue de resultados, la finalidad del módulo Protocolo RS232 es poder establecer una interfaz entre el sistema manipulador de video autónomo y una PC. Esta interfaz permite al usuario poder visualizar los resultados de adquisición y manipulación de la señal de video, directamente en la PC permitiendo una evaluación de los resultados del sistema.

Este módulo está formado por las secciones, de transmisión y recepción, descritas a continuación.

La recepción de datos por parte del FPGA tiene como función primordial recibir comandos provenientes desde la PC, tales como el envío de una imagen,

ruido y sobre muestreo, todo esto hace posible que se utilice en aplicaciones de telecomunicaciones, instrumentación y video.

El módulo también controla el acceso al bloque de memoria de 1024 Kbytes que contiene el sistema, este bloque de memoria esta formado por 2 memorias SRAM perteneciente a la familia KM684000B de SAMSUNG. En este módulo de memoria se almacenan las imágenes adquiridas y los resultados de los algoritmos que se les aplican, para su manipulación y/o interpretación.

Los protocolos anteriores están en función de las señales provenientes de la etapa analógica, la cual se encarga de adecuar la señal de video para su adquisición. El módulo de control y sincronía se encarga de detectar las señales de sincronía del video para digitalizar la información de éste.

Las figuras 5 y 6 muestran la simulación de los procesos de detección de las señales de sincronía y almacenamiento de un píxel de una imagen respectivamente, dentro del Módulo de Control y sincronía.

iniciar un nuevo proceso de digitalización, etc.

La recepción de un dato da inicio cuando las señales **RXActive** y **RX** son iguales a 0, indicando con esto que la PC ha empezado a enviar un bit de inicio, esto se puede apreciar en la figura 7.

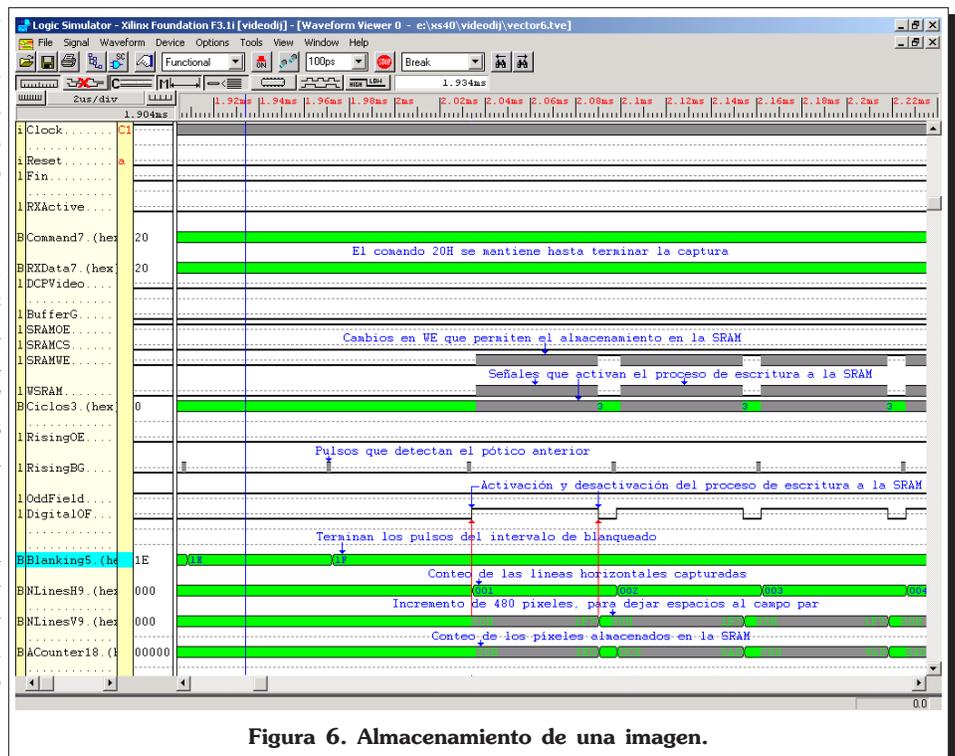


Figura 6. Almacenamiento de una imagen.

La siguiente condición se cumple cuando **RXActive** se ha activado, **RXDelay** está desactivado ya que no ha transcurrido el tiempo de espera e **iPR1** es igual 156. Este valor corresponde a la mitad del tiempo que dura un bit en una transmisión con velocidad de 115200 baudios, la forma de obtener este valor es dividiendo la frecuencia del reloj maestro entre la velocidad de transmisión deseada y luego dividirlo entre 2 ya que corresponde a la mitad de un bit como se muestra en la ecuación 1. La condición termina activando a **RXDelay** y asignando valores iniciales a los objetos **iPR1** e **iPR2**.

$$\frac{\text{Frecuencia de Operación del Sistema}}{\text{(2) Velocidad de Transmisión}} = \frac{\text{Ciclos de retardo}}{\dots\dots\dots} \text{ Ec. (1)}$$

Cuando se ha activado la recepción de un carácter y ha transcurrido el tiempo de retardo, la posición que se ha alcanzado es la mitad del tiempo que dura el bit de inicio, es necesario esperar 312 ciclos de reloj más para almacenar la muestra del bit menos significativo, después de transcurrir los siguientes 312 ciclos es posible tomar la muestra del segundo bit y así sucesivamente. Los 8 bits se almacenan en **RXData** teniendo en su localidad 0 el bit LSB y en su localidad 7 el bit MSB, después de que se han capturado los 8 bits el proceso de recepción espera aproximadamente 370ms aunque podría esperarse menos tiempo, sin embargo aquí se ha utilizado este valor

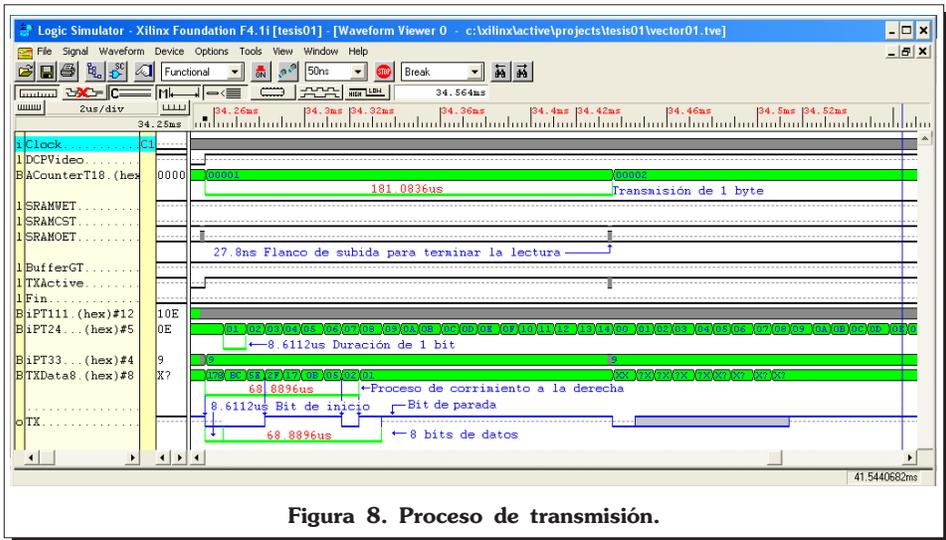


Figura 8. Proceso de transmisión.

módulo de transmisión recupera el primer píxel de la SRAM y lo envía hacia la PC vía puerto serie, este proceso se repetirá hasta terminar de enviar todos los pixeles que componen una imagen, **figura 8**. El número de pixeles que compone una imagen está en función del registro configurable **Counter1** y del reloj del convertidor analógico digital, **tabla 1**.

Valor de Counter1	Frecuencia de reloj del ADC (MHz)	Resolución horizontal de la imagen
2	9.0	495
3	6.0	330
4	4.5	247
5	3.6	198

Tabla 1. Parámetros de resolución de una imagen.

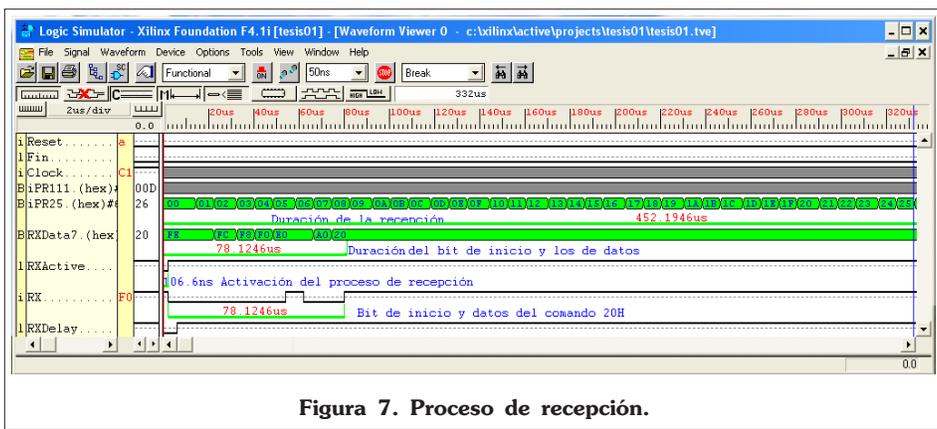


Figura 7. Proceso de recepción.

para asegurar el dato, como se ilustra en la **figura 7**. El proceso de transmisión se inicia con una orden de la PC con la finalidad que el sistema autónomo de adquisición de imágenes le envíe la última imagen digitalizada. El

MÓDULO DE MANIPULACIÓN DE VIDEO

Se pretende que una vez adquirida la señal de video se puedan implementar algoritmos de procesamiento de imágenes, tales como gradientes, detección de bordes, filtrado de la señal, codificación, etc.

Este módulo reserva recursos dentro del FPGA para poder implementar ahí los algoritmos de procesamiento de imágenes.

La **figura 9** muestra el resultado de la adquisición de una imagen.

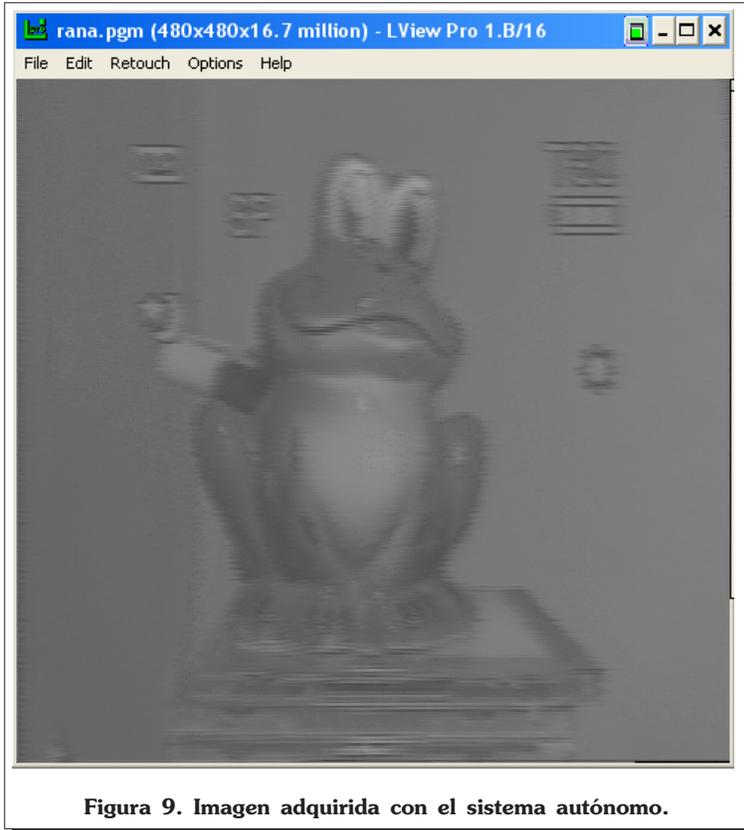


Figura 9. Imagen adquirida con el sistema autónomo.

APLICACIONES

En este momento se está haciendo uso de este sistema en las siguientes aplicaciones:

Visión artificial para móviles. Dotar de visión a un robot mediante la implementación de algoritmos de procesamiento de imágenes tales como detección de bordes y gradientes.

Sistema autónomo para reconocimiento de rostros. Utilizarlo en sistemas de seguridad para permitir el acceso con base al reconocimiento de ciertas características del rostro humano.

CONCLUSIONES

El resultado de este trabajo es un sistema que incluye las ventajas inherentes de utilizar FPGA's, como son la versatilidad, alta velocidad de procesamiento, fácil adaptación del sistema, flexibilidad, tiempo de diseño, etc., características necesarias en aplicaciones de tratamiento de imágenes.

La reconfiguración es una característica más que un FPGA ofrece, con ella se puede lograr que este sistema

pueda implementar diversos algoritmos, en distintos tiempos, mediante los mismos recursos de hardware.

Al emplear un lenguaje descriptor de hardware como VHDL, permite que el diseño posea una característica importante para el mantenimiento del sistema, la modularidad, esto implica que el diseño está dividido en varias secciones, lo cual facilita hacer un rediseño del sistema o incluir secciones nuevas de propósito específico.

Las áreas donde se puede aplicar este sistema son amplias, visión artificial para móviles, sistema autónomo para reconocimiento de rostros, reconocimiento de huellas digitales, codificadores de video o imágenes tales como JPEG (Joint Photographic Experts Group), etc.

Con cambios pequeños, el digitalizador podrá trabajar con los estándares de video a color PAL o SECAM, soportando el formato NTSC por default.

REFERENCIAS

- [1] F. Blasco, F. Pardo, J. A. Boluda, "Sistema de adquisición de imágenes Log-Polares con alta velocidad basada en bus PCI", Dpt de Informática y Electrónica, Universidad de Valencia, España, 1999.
- [2] J. C. Vélez, A. Echeverría, A. Gallón, "CHIP para adquisición y edición de video", Grupo de Microelectrónica, U.P.B., Medellín, Colombia, 2000.
- [3] E. Mandado, L. J. Álvarez, D. Valdés, "Dispositivos Lógicos Programables y sus aplicaciones", Ed. Thomson, España, 2002, pp. 76-99, pp.130-134.
- [4] *The programmable Logic Data Book 2000*; Xilinx, Inc., 2000, CD-ROM.
- [5] S. A. Pérez, E. Soto, S. Fernández, "Diseño de Sistemas Digitales con VHDL", Ed. Thomson, España, 2002.
- [6] D. Pellerin, D. Taylor, "VHDL, Made Easy!", Ed. Prentice-Hall, USA, 1997.
- [7] *IEEE standard VHDL Language Reference Manual*, IEEE Standard 1076-1993, 1994.
- [8] Craig A. Lindley, "Practical image processing in C", Ed. John Wiley & Sons, USA, 1991, pp. 60-64.

[URL1] Texas Instruments. <http://dspvillage.ti.com>

[URL2] Xilinx, Inc. <http://www.xilinx.com>

[URL3] Xess, Corp. <http://www.xess.com>

Diseño de Control Difuso Usando Promedio de Pesos e Implementado con Lenguaje Verilog

Romeo Urbietta Parrazales
rurbietta@cic.ipn.mx
 Profesor del CIC-IPN

Enrique Guzmán Ramírez
eguzman@mixteco.utm.mx
 Instituto de Electrónica y Computación,
 Universidad Tecnológica de la Mixteca

Los controladores que son tradicionalmente implementados en microcontroladores solamente pueden incorporar superficies de control abruptas. El propósito de este trabajo es implementar un control difuso con superficies mas finas dentro de un FPGA. El FPGA ha permitido a los diseñadores de control crear mejores diseños de prueba, hacer modificaciones muy fáciles y rápidas, en un espacio de dimensiones muy reducidas e inmunes al ruido. Esta aproximación de control difuso usa un nuevo concepto de promedio de pesos para tratar de disminuir la tabla de valores difusos, aunque los tamaños de la entrada sean grandes. El hardware del prototipo se implementó a través de un sistema de desarrollo que contiene un FPGA de la familia Spartan IIE tipo PCI y su memoria PROM. Esta fue acoplada por un conector paralelo e interfaz JTAG a la PC, y por el otro lado a un módulo analógico P160. La implementación de software del nuevo algoritmo de control se forma por tres rutinas en lenguaje Verilog. La implementación de software de la superfi-

cie de control a suavizar se instaló en la PROM. El resultado final fue un sistema de control difuso con cuatro entradas de siete bits cada una y una salida de dieciséis bits.

I. INTRODUCCIÓN

El diseño de controladores difusos basados en FPGAs puede ser muy simple, consistiendo de un FPGA, dos convertidores analógico-digital (A/D), un convertidor digital digital-analógico (D/A) para la salida y un chip ROM. El chip ROM se usa para la tabla de búsqueda difusa (LUT). La razón de tener la LUT en un chip externo es para evitar que el FPGA sea reprogramado si la superficie de control cambia, solamente el “chip” ROM externo necesitará ser cambiado. El diagrama a bloques de un controlador difuso basado en FPGAs se ilustra en la **figura 1**.

La implementación de un sistema difuso tradicional es fácilmente programada dentro de un FPGA y trabaja bien para dos entradas[1]. Sin embargo, si las entradas son incrementadas a tres, la TB (tabla de búsqueda = LUT), usando el método de implementación tradicional, llega a ser muy difícil de manejar.

El tamaño del LUT también crece exponencialmente tanto como entradas sean agregadas[2], siendo este uno de los problemas del control difuso. Se han hecho intentos para corregir el problema al combinar dos o mas entradas para el controlador difuso, tal que el número de entradas permanecerá siempre alrededor dos.

II. DISEÑO DIGITAL DEL ALGORITMO PROMEDIO DE PESOS

ARQUITECTURA INTERNA DEL NUEVO ALGORITMO DE CONTROL

La aproximación presentada usa el concepto de promedio de pesos para conservar en forma pequeña el LUT difuso, cuando el tamaño de la entrada se hace grande. Esto se lleva a cabo usando los bits 3 y 4 más significativos de cada entrada para determinar la dirección del LUT[3]. El promedio de pesos elimina los bits

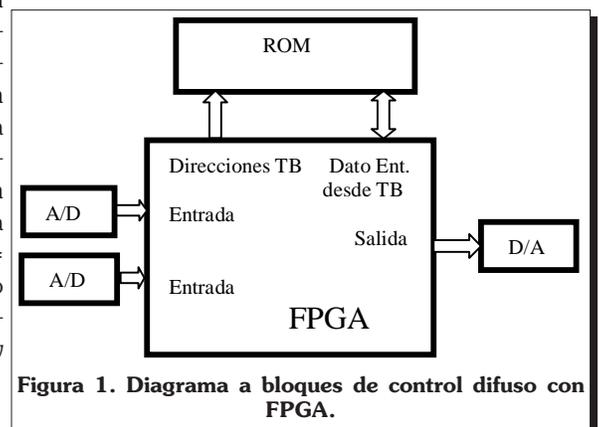
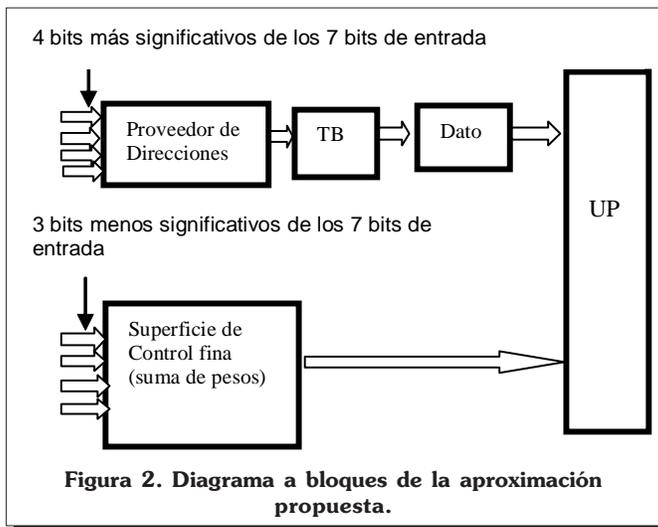


Figura 1. Diagrama a bloques de control difuso con FPGA.



PROVEEDOR DE DIRECCIONES DE LA TABLA DE BÚSQUEDA

Con el concepto de promedio de pesos, la siguiente preocupación fue determinar si los ejes de esta superficie de control acoplarían las superficies de control vecinas. El bloque de direcciones [4] de la **figura 2** puede ser visto ahora en la **figura 3**.

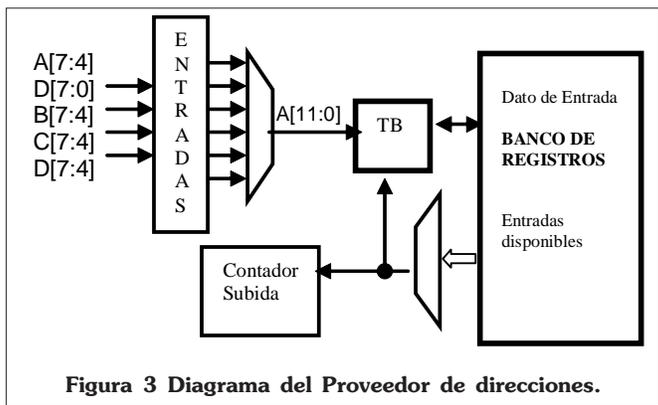
remanentes para eliminar lo escarpado o abrupto de la superficie de control. El diagrama a bloques del sistema se muestra en la **figura 2**. Este nuevo método puede causar un decrecimiento en la velocidad (la cual es no crucial si se implementa en un FPGA) pero se obtiene una superficie suave, y además se puede diseñar para un gran número de entradas. En este proyecto se planteó para cuatro.

Como se ve en esta figura, hay cuatro entradas. Esto es una buena prueba para determinar si la aproximación promedio de pesos trabaja con más de dos entradas. La parte difícil del proyecto es la implementación de la nueva aproximación de suma de pesos.

Este diagrama a bloques tiene un multiplexor grande para las entradas. Para esto se empleó el software de Xilinx, que sería capaz de ubicarlo y rutearlo eficientemente. El diseño de la **figura 4** fue creado para implementar eficientemente el multiplexor, al dividirlo en cuatro multiplexores más pequeños.

PROMEDIO DE PESOS

Con esta parte del diseño, la siguiente meta fue diseñar la parte del promedio de pesos, así, aquellos valores podrían ser multiplicados por la tabla y entonces acumulados para la respuesta. El diagrama a bloques del promedio de pesos está en la **figura 5**. Aquí, los bloques de entrada son los bits menos significativos, los cuales son invertidos o pasados a través de un patrón específico.



UNIDAD DE PROCESAMIENTO

El propósito del bloque de procesamiento es multiplicar los valores promedio de los pesos propios por los valores de la tabla de búsqueda TB (LUT). El proceso entonces suma todos los

Figura 4. Diagrama a Bloques para la Estructura de Multiplexores de Entrada.

valores para crear la salida. El primer diseño del bloque de procesamiento intentado fue de 16 multiplicadores paralelos seguido por 15 sumadores. Esto causó que el diseño fuera muy grande, lo cual no es muy recomendable. La segunda aproximación de diseño al bloque de procesamiento fue hecho en una TB, tomando los dos valores de entrada y concatenándolos juntos para crear una dirección de TB construida.

Esta aproximación de diseño no es muy recomendable tampoco. La tercera aproximación de diseño fue para reemplazar los multiplicadores en paralelo con un simple multiplicador y usarlo secuencialmente. El diagrama a bloques de esta aproximación a la unidad de procesamiento puede ser visto en la **Figura 6**.

Los recursos compartidos eliminan muchas compuertas. Otra aproximación que debería considerarse es posicionar algunos de los sumadores a lo largo de los multiplicadores, si el espacio fuera todavía suficiente.

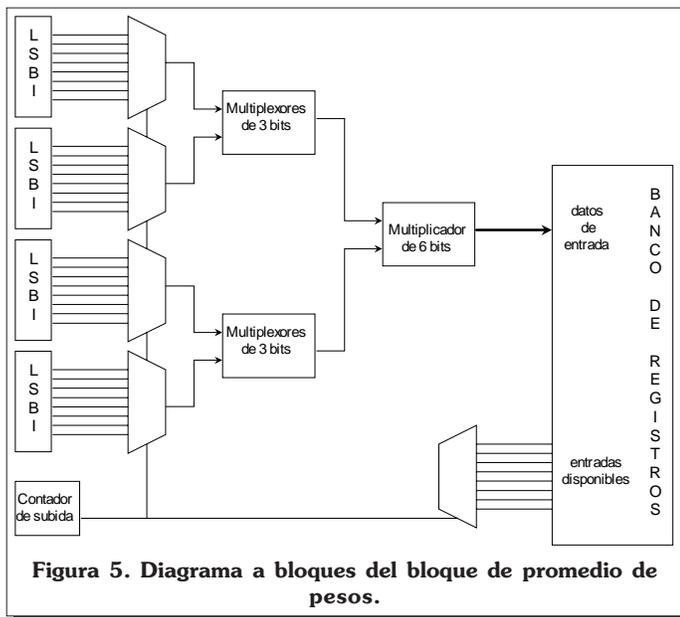


Figura 5. Diagrama a bloques del bloque de promedio de pesos.

III. IMPLEMENTACION DEL ALGORITMO DE PROMEDIO DE PESOS EN VERILOG (FPGA)

DISEÑO FINAL DE CONTROL

Este diseño necesita 68 patas disponibles de Entrada/Salida. Hay cuatro entradas de 7 bits, dos líneas de salida de 16 bits y una entrada más de 8 bits. Las cuatro entradas de 7 bits son entradas para el diseño. La primera salida de 16 bits es la línea de direccionamiento para la tabla de búsqueda y la entrada de 8 bits es la línea de datos desde la tabla de búsqueda. La salida final de 16 bits es la salida del diseño. Un diagrama a bloques

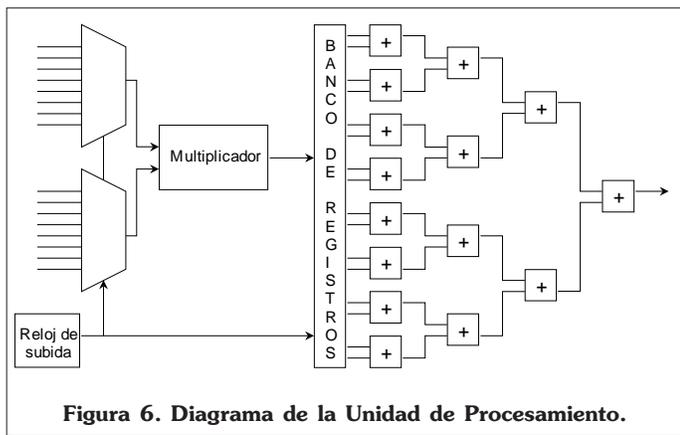


Figura 6. Diagrama de la Unidad de Procesamiento.

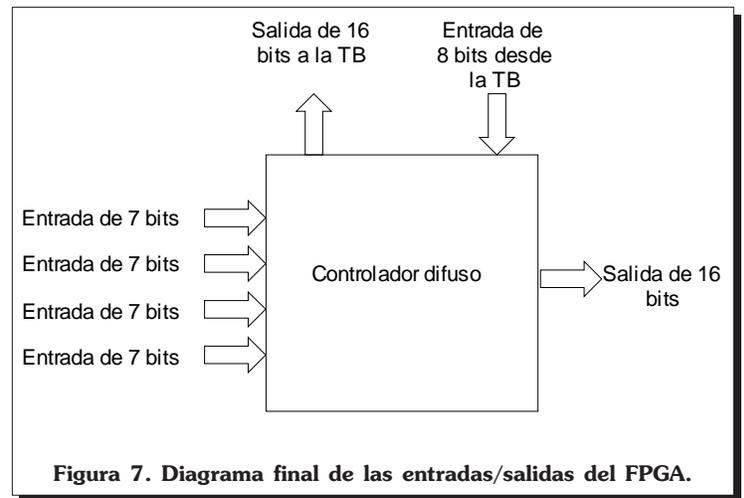


Figura 7. Diagrama final de las entradas/salidas del FPGA.

ques de la estructura de entrada/salida se puede ver en la figura 7. El chip seleccionado para el diseño es el XC2S50E de 144 patas[5], con tecnología de montaje de superficie.

CONCLUSIONES

El diseño trabaja muy bien y las salidas son muy suaves. El nuevo método de promedio de pesos es muy superior a los métodos tradicionales de control difuso. El sistema tradicional provee unas superficies rugosas y abruptas, los cuales pueden causar que el sistema ha controlar se vuelva inestable. Con superficies más suaves el sistema controlado por el nuevo controlador difuso de promedio de pesos es más estable en comparación con los métodos tradicionales. En sistemas difusos tradicionales el número de entradas son usualmente de dos a tres entradas. Este método permitió un número de entradas más

grandes, en este caso fueron cuatro entradas. El método de promedio de pesos ayuda a resolver los problemas de crecimiento exponencial y complejidad de la tabla de búsqueda, permitiendo de esta manera superficies más suaves con pequeñas tablas de búsqueda.

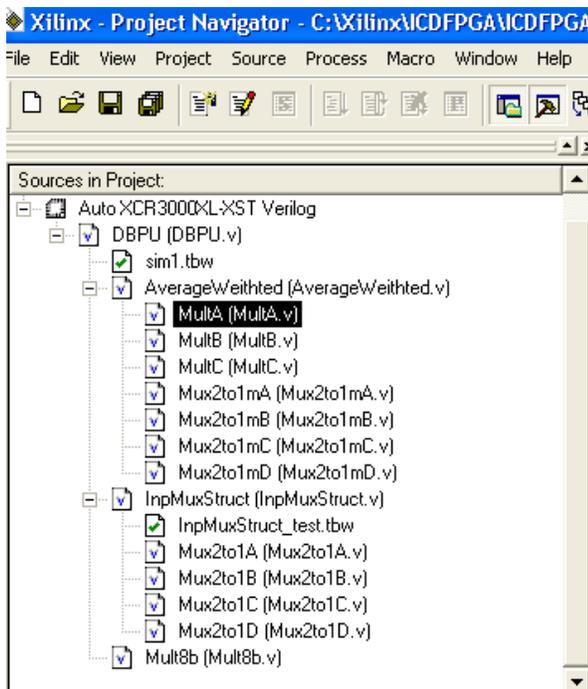
Esta nueva tecnología de sistemas digitales es de bajo consumo de voltaje y corriente y de frecuencias muy elevadas con respecto de aquellos circuitos integrados rígidos (hechos en la fábrica), lo que lo hacen inmunes al ruido. La nueva tecnología empleada se podría pensar en similitud con aquellas personas que envían a ser un traje a la medida, así, estos circuitos pueden incorporar algoritmos de control difusos más sencillos hasta los más complejos en una forma específica que no se encuentra en el mercado.

REFERENCIAS

- [1] Pedro Diniz and Joonseok Park. "Automatic Synthesis Data Storage and Control Structures for FPGA-based Computing Engines". University of Southern California.
- [2] S K Kaul, R Koul, C L Bhat, I K Kaul and A K Tickoo. "Use of a 'look-up' table improves the accuracy of a low-cost resolver-based absolute shaft encoder". Bhabha Atomic Research Centre, Nuclear Research Laboratory, Trombay, Mumbai 400 085, India. December 1996.
- [3] Chr. W. Frey, A. Jacobasch, H.-B. Kuntze, R. Plietsch. "Smart Neuro-Fuzzy Based Control of a Rotary Hammer Drill". Fraunhofer Institute for Information and Data Processing IITB. Fraunhoferstraße 1, D-76131 Karlsruhe, Germany.
- [4] John M. Yarbrough. "Digital Logia". Decoders. PWS Publishing Company. Pag. 171-184. 1997.
- [5] Spartan II 200 PCI Development Board User's Guide. Version 1.0 April 2002.

ANEXO I

PROGRAMA PRINCIPAL DE CONTROL DIFUSO USANDO PROMEDIO DE PESOS EN LENGUAJE VERILOG.



SUBROUTINAS DE PROMEDIO DE PESOS

```

module MultA (OutmA, OutmB, MAB);
    input  [2:0] OutmA, OutmB;
    output [5:0] MAB;

    assign MAB = OutmA * OutmB;

endmodule
    
```

```

module MultC (MAB, MCD, MABCD);
    input  [5:0] MAB, MCD;
    output [11:0] MABCD;

    assign MABCD = MAB * MCD;

endmodule
    
```

```

module MultB (OutmC, OutmD, MCD);
    input  [2:0] OutmC, OutmD;
    output [5:0] MCD;

    assign MCD = OutmC * OutmD;

endmodule
    
```

```

module Mux2to1mA(A, OutmA, Counter);

    input  [1:0] Counter; // Se g
    input  [7:0] A;
    output [2:0] OutmA;

    reg   [2:0] OutmA;

    always @(Counter or A)
        assign OutmA = A[3:1]; //

endmodule
    
```

```

module Mux2to1mB(B, OutmB, Counter);

    input  [1:0] Counter; // Se
    input  [7:0] B;
    output [2:0] OutmB;

    reg   [2:0] OutmB;

    always @(Counter or B)
        assign OutmB = B[3:1];

endmodule
    
```

```

module Mux2to1mC(C, OutmC, Counter);
    input  [1:0] Counter;    // Se
    input  [7:0] C;
    output [2:0] OutmC;

    reg    [2:0] OutmC;

    always @(Counter or C)
        assign OutmC = C[3:1];
endmodule
    
```

```

module Mux2to1B(B, OutB, Counter);
    input  [1:0] Counter;    // Se
    input  [7:0] B;
    output [7:0] OutB;

    reg    [7:0] OutB, AuxB;

    always @(Counter or B or AuxB)
    begin
        assign AuxB = B[7:4] + 1;

        case (Counter)
            2'd0 : OutB = AuxB; // ADD
            2'd1 : OutB = B[7:4];
        endcase
    end // Fin del always
endmodule
    
```

```

module Mux2to1mD(D, OutmD, Counter);
    input  [1:0] Counter;    // Se
    input  [7:0] D;
    output [2:0] OutmD;

    reg    [2:0] OutmD;

    always @(Counter or D)
        assign OutmD = D[3:1];
endmodule
    
```

```

module Mux2to1C(C, OutC, Counter);
    input  [1:0] Counter;    // Se
    input  [7:0] C;
    output [7:0] OutC;

    reg    [7:0] OutC, AuxC;

    always @(Counter or C or AuxC)
    begin
        assign AuxC = C[7:4] + 1;

        case (Counter)
            2'd0 : OutC = AuxC; // ADD
            2'd1 : OutC = C[7:4];
        endcase
    end // Fin del always
endmodule
    
```

SUBROUTINAS DE DIRECCIONAMIENTO DE LA PROM

```

module Mux2to1A(A, OutA, Counter);
    input  [1:0] Counter;    // Se
    input  [7:0] A;
    output [7:0] OutA;

    reg    [7:0] OutA, AuxA;

    always @(Counter or A or AuxA)
    begin
        assign AuxA = A[7:4] + 1;

        case (Counter)
            2'd0 : OutA = AuxA; // ADD
            2'd1 : OutA = A[7:4];
        endcase
    end // Fin del always
endmodule
    
```

```

module Mux2to1D(D, OutD, Counter);
    input  [1:0] Counter;    // Se
    input  [7:0] D;
    output [7:0] OutD;

    reg    [7:0] OutD, AuxD;

    always @(Counter or D or AuxD)
    begin
        assign AuxD = D[7:4] + 1;

        case (Counter)
            2'd0 : OutD = AuxD; // ADD
            2'd1 : OutD = D[7:4];
        endcase
    end // Fin del always
endmodule
    
```

Arquitectura Básica para Controladores de Lógica Difusa a Programarse en FPGAs

Juan C. Herrera Lozada,
 jcrls@ipn.mx
 Ma. de Lourdes Olvera Cárdenas,
 lolvera@ipn.mx
 Ma. Teresa Lozano Hernández,
 tlozanoh@ipn.mx
 Profesores del CIDETEC-IPN

Este artículo muestra el diseño de una arquitectura básica de un controlador de lógica difusa (FLC – Fuzzy Logic Controller) con modelo orientado a cálculos, a implementarse en un FPGA. La descripción del funcionamiento se realizó alternando entre VHDL y Verilog.

1. INTRODUCCIÓN

Las técnicas de la lógica difusa son utilizadas constantemente para resolver problemas no lineales de control, debido a que toma múltiples valores, causando una transición gradual de uno a otro.

El proceso de lógica difusa se divide en tres interfaces, cuya participación en el proceso no cambia de orden: Fuzificación, Inferencia y Defuzificación. En la **Figura 1** se advierte que la Inferencia está supeditada por los valores que entrega la Fuzificación; así mismo, la Defuzificación depende de los propios de la inferencia. Por lo anterior, se afirma que el proceso difuso es secuencial por naturaleza; sin embargo, existe la posibilidad de paralelizar las tareas internas

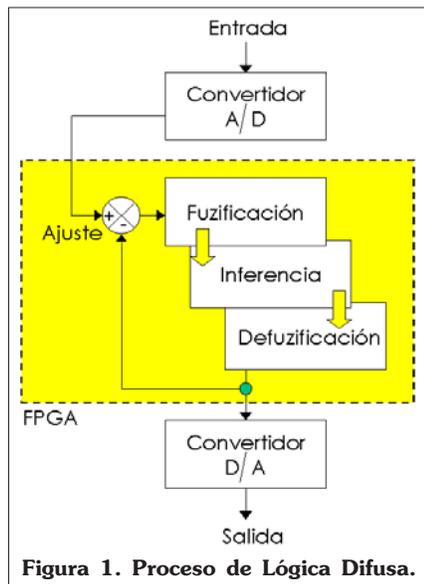


Figura 1. Proceso de Lógica Difusa.

que realiza cada una de las interfaces, propiciando una respuesta más aventajada con respecto a la velocidad de cálculos.

Proponiendo una división modular intrínseca a cada interfaz es posible definir una arquitectura que sea adaptable a cualquier sistema sin importar el campo de diligencia, haciendo extensiva la metodología de diseño.

Los controladores de naturaleza digital presentan arquitecturas determinadas por la forma en la que obtienen sus datos para comenzar las operaciones especializadas: si provienen de tablas con valores de fuzificación y defuzificación precalculados, o si se obtienen en el mismo instante con ayuda de unidades aritméticas en hardware. En este sentido se tienen

dos tipos de modelos a seguir: *Modelo Orientado a Memoria* y *Modelo Orientado a Cálculos*.

El *Modelo Orientado a Memoria*, también llamado *Modelo de Obtención Indirecta de Datos Difusos*, presenta la particularidad de utilizar tablas precalculadas de valores, almacenadas en una memoria que puede ser externa o interna al controlador. Los datos pueden ser calculados directamente por el diseñador o como en la mayoría de los casos (incluidas las versiones comerciales) se utiliza un software que realiza los cálculos y crea el ensamblador que configura las características que cumplirá el controlador de acuerdo a los requerimientos propuestos por el diseñador. La **Figura 2** esquematiza un controlador orientado a memoria.

La utilización de un dispositivo de memoria para almacenar las tablas precalculadas implica el diseño de un controlador de memoria y a la vez de un sistema de sincronización por ciclos para la obtención de cada valor.

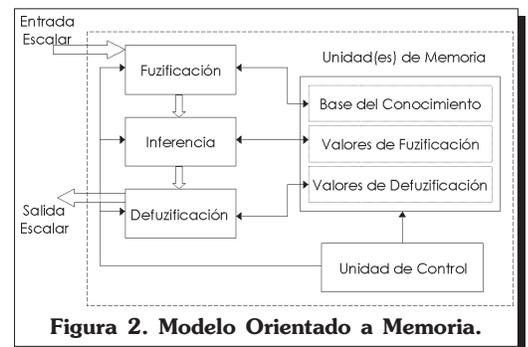


Figura 2. Modelo Orientado a Memoria.

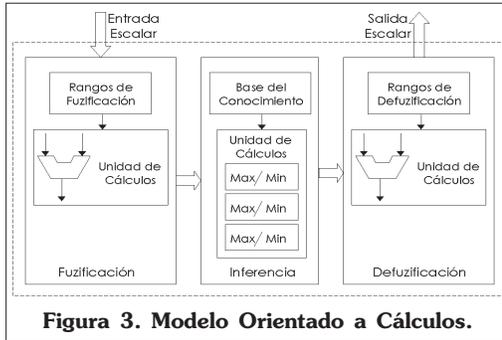


Figura 3. Modelo Orientado a Cálculos.

La velocidad de acceso para lectura y escritura, es una limitante física en este tipo de modelos; aunado a que los datos se van obteniendo uno a uno a la vez. Para sistemas mecánicos simples, este tipo de modelo es una buena opción dado que la gran mayoría de los controladores difusos dedicados se diseñan de acuerdo a esta orientación, así el mercado actual contempla un soporte muy importante en lo concerniente al software y dispositivos de propietario.

La **Figura 3** muestra una aproximación a la arquitectura de un *Modelo Orientado a Cálculos*. También se conoce como *Modelo de Obtención Directa de Datos Difusos* y tiene la característica de utilizar unidades aritméticas construidas en software o hardware, que realizan el cálculo a medida que se tiene un valor a procesar. La estimación de las arquitecturas orientadas a este modelo necesitan de algoritmos aritméticos óptimos en la implementación de cada una de las unidades de cálculo. Este modelo no requiere de cálculos previos y específicamente es el tipo de orientación que se utilizó en este documento.

2. DISEÑO PLANTEADO

Para unificar criterios, se premedita el ejercicio de un *controlador de irrigación*. Se propone que una *válvula de irrigación* se controle con dos variables de entrada: *Temperatura del Aire* y

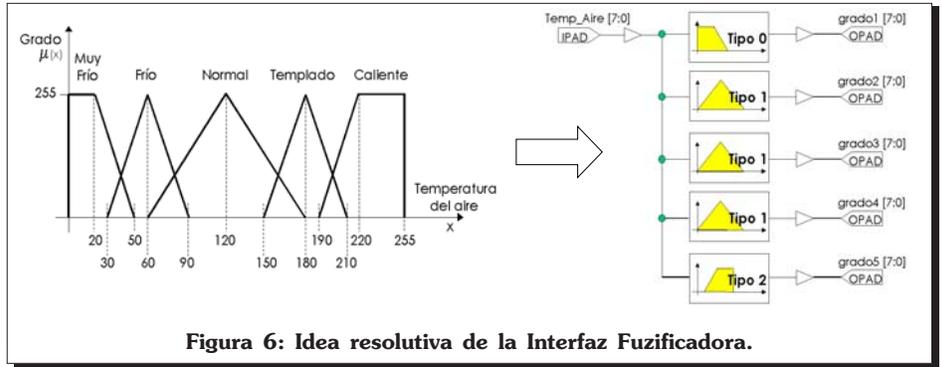


Figura 6: Idea resolutiva de la Interfaz Fuzificadora.

Humedad de la Tierra. El FLC deberá hacer que la combinación de ambas variables determine el *Tiempo de Irrigación*. Las unidades de medida de cada variable se consideran normalizadas en un rango de 0 a 255, para un controlador de 8 bits, sin signo y con escalares de punto fijo. Para la realización hardware, se trabaja con diseños independientes para cada interfaz y posteriormente se integran para conformar el controlador completo.

entrada real en valores difusos (grados de pertenencia), utilizando la ecuación de la línea recta.

Una variable a fuzificar con n funciones de pertenencia diferentes se resuelve paralelamente con n diferentes bloques fuzificadores, donde cada uno calcula de manera autónoma su grado de pertenencia, tal y como se aprecia en la **Figura 6**.

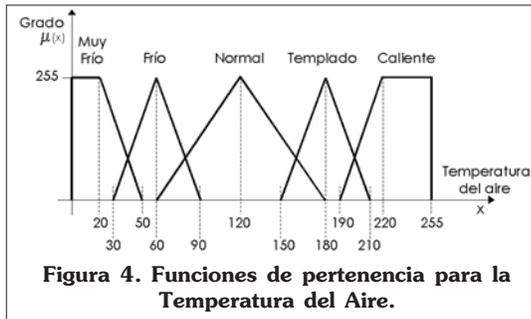


Figura 4. Funciones de pertenencia para la Temperatura del Aire.

2.1 DISEÑO DE LA INTERFAZ DE FUZIFICACIÓN

Las **Figuras 4** y **5**, muestran las funciones de pertenencia para la fuzificación de ambas variables de entrada. Esta interfaz, debe convertir la

Cada bloque fuzificador tiene puntos gráficos característicos que se modifican de manera simple en código. Supóngase la función de pertenencia *Frío* de la **Figura 4**. Una aproximación al código en Verilog que modela su funcionamiento se lista a continuación.

```

always @(Temp_Aire)
begin
pendiente = 255/30;
if (Temp_Aire<=30 || Temp_Aire>=90)
grado = 0;
else if (Temp_Aire<=60)
grado = pendiente*(Temp_Aire - 30);
else
grado= pendiente*(90 - Temp_Aire);
end
    
```

El formato textual y comprensible de la sintaxis permite modelar funciones de pertenencia de los tipos habituales (triangulares y trapezoidales) con pendientes simétricas y asimétricas, a excepción del Gaussiano debido a que su solución se apega más a medios analógicos que digitales. La **Figura 7** (siguiente página), presenta el algoritmo en hardware, codificado anteriormente.

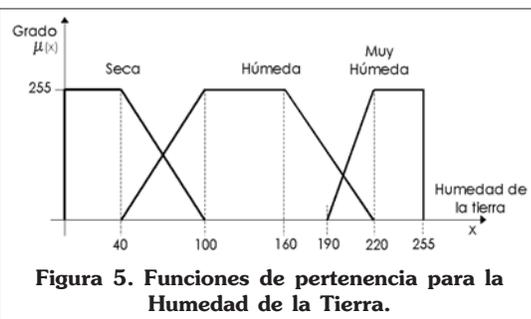


Figura 5. Funciones de pertenencia para la Humedad de la Tierra.

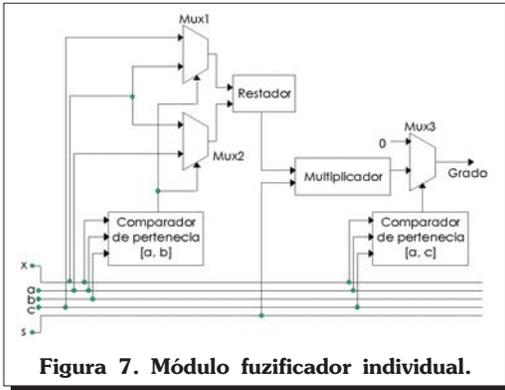


Figura 7. Módulo fuzificador individual.

2.2. DISEÑO DE LA INTERFAZ DE INFERENCIA

Para modelar la interfaz de Inferencia se propone la solución Mamdani MAX-MIN, donde las reglas individuales se evalúan con el operador del mínimo de ambos grados y si comparten el consecuente, se evalúan con el máximo. Una idea abstracta se muestra en la **Figura 8**, pensando que cada una de las dos unidades fuzificadoras planteadas de inicio, aporta su respectivo grado calculado. Las reglas se evalúan todas contra todas en un esquema matricial, como lo indica la **Tabla 1**.

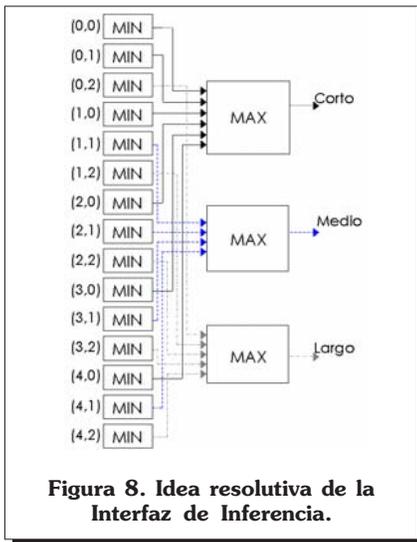


Figura 8. Idea resolutive de la Interfaz de Inferencia.

Cada regla se conecta a su respectivo consecuente de manera personalizada, con la ventaja que los bloques MIN y MAX, no cambian de estrategia en la codificación. La **Figura 9**, representa un bloque individual.

		Temperatura del Aire				
		Muy Frio	Frio	Normal	Templado	Caliente
Humedad de la Tierra	Muy Húmeda	Corto	Corto	Corto	Corto	Corto
	Húmeda	Corto	Medio	Medio	Medio	Medio
	Seca	Largo	Largo	Largo	Largo	Largo

Tabla 1. Reglas de Inferencia.

Una aproximación a la codificación del bloque anterior en VHDL se lista a continuación. Para describir un bloque MIN sólo basta modificar la comparación.

```

process (A,B)
begin
if (A<B) then
consecuente <= A;
else
consecuente <= B;
end if;
end process;
    
```

2.3. DISEÑO DE LA INTERFAZ DE DEFUZIFICACIÓN

Suponiendo las tres funciones Singletons de defuzificación planteadas en la **Figura 10**, se procede a utilizar el método del Promedio de los Centros para entregar el valor real.

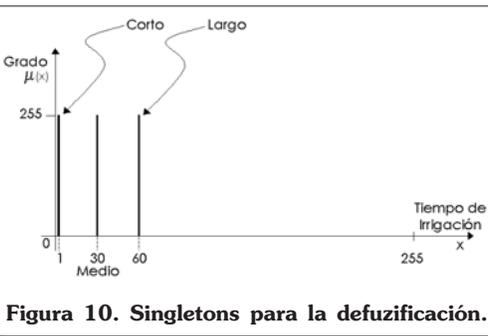


Figura 10. Singletons para la defuzificación.

La **Figura 11**, muestra la modularización del proceso defuzificador. Los consecuentes entregados por la interfaz de Inferencia (Figura 7) se multiplicarán de forma individual con su respectivo peso (valor Singleton).

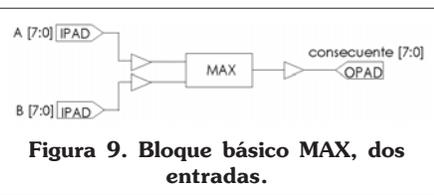


Figura 9. Bloque básico MAX, dos entradas.

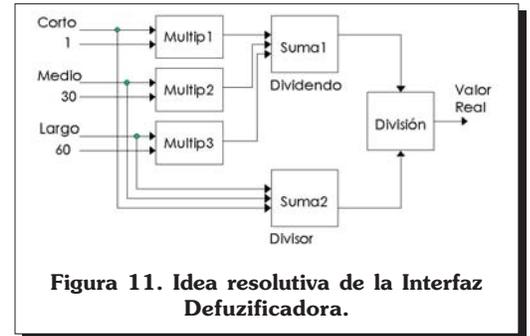


Figura 11. Idea resolutive de la Interfaz Defuzificadora.

Describir un multiplicador en VHDL o Verilog es elemental tal y como lo muestra el listado siguiente en VHDL. Es importante mantener el tamaño del resultado.

```

Process(A, B)
prod <= Conv_Std_Logic_Vector(A*B),16);
End Process;
    
```

Una aproximación al listado de un sumador en Verilog se lista parcialmente a continuación.

```

always @ (A or B or C)
begin
divisor = A + B + C;
end
    
```

Para la unidad de división por hardware, se optó por realizar una unidad síncrona y otro más asíncrona. La primera realiza restas sucesivas, una por cada ciclo de reloj, por lo que si se tiene un dividendo de 255 y un divisor de 4, tendrán que pasar 60 ciclos de reloj para visualizar el resultado. El listado parcial, en la siguiente página) muestra una aproximación a la codificación de este algoritmo de división, haciendo hincapié en el ciclo While que realiza las restas sucesivas.

Opción 1	Opción 2
<pre> always @(dividendo or divisor) begin : divide cont = 0; minuendo = dividendo; sustraendo = divisor; if (!minuendo !sustraendo) begin resul = 0; disable divide; end else if (minuendo == sustraendo) begin resul = 1; disable divide; end else begin while (minuendo >= sustraendo) begin resta = minuendo - sustraendo; cont = cont + 1; minuendo = resta; end resul = cont; disable divide; end end endmodule </pre>	<pre> always @(dividendo or divisor) begin : divide integer i; reg_cor=(17'b0, dividendo); c_divisor=(9'b0, divisor); for (i=0; i<16; i=i+1) begin reg_cor= reg_cor<<1; reg_cor[32:16]= reg_cor[32:16]-c_divisor; guarda[15-i]= reg_cor[32]; if (reg_cor[32]) reg_cor[32:16]=reg_cor[32:16]+c_divisor; resul=guarda[7:0]; end disable divide; end </pre>

En la segunda opción, se utiliza el algoritmo de corrimientos constantes que permite obtener el dato de manera inmediata.

3. INTEGRACIÓN E IMPLEMENTACIÓN

La implementación del controlador se realizó sobre un FPGA XC4010XL de Xilinx con resultados de cálculo satisfactorios. La **Figura 12** representa una aproximación al diagrama esquemático, indicando

puntualmente la modularización del diseño.

La **Figura 13** muestra la tarjeta conectada con sus interfaces de entrada y salida. Para introducir los datos y validar resultados se utilizaron ADCs e interruptores convencionales y una pantalla LCD genérica. La unidad adquisitiva de datos, así como los acoplamientos exteriores del FLC, no se consideran elementos de discusión en los alcances de este trabajo.

Los códigos diseñados se sintetizaron con ayuda del software Founda-

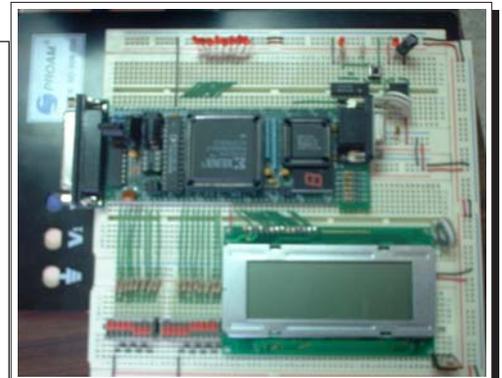


Figura 13. Tarjeta con interfaces de entrada y salida.



Figura 14. comprobación Experimental

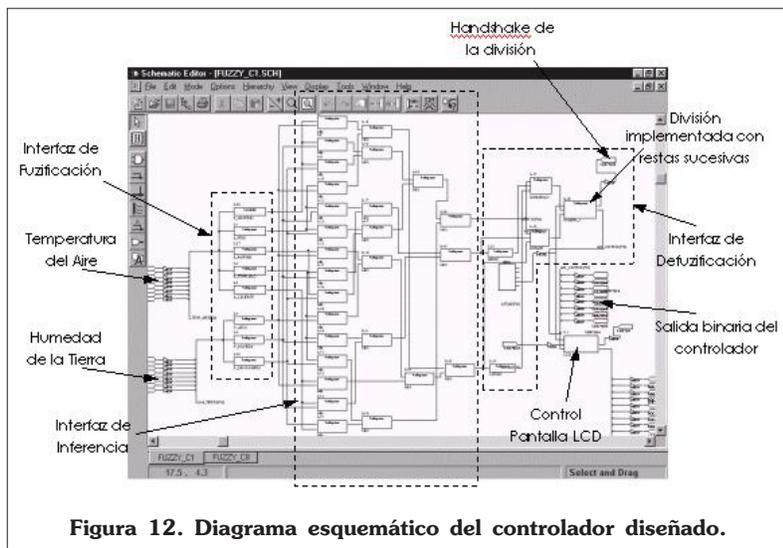


Figura 12. Diagrama esquemático del controlador diseñado.

tion de Xilinx en su versión 4.1i, obteniéndose 331 CLBs (Módulos Lógicos) para el controlador con división síncrona y 840 CLBs para el propio de la división asíncrona. La frecuencia de trabajo es de hasta 6MHz, siendo suficientes 16 KHz para elementos de respuesta mecánica.

La **Figura 14** muestra una validación de resultados, comprobados experimentalmente.

4. CONCLUSIONES

Se logró implementar un controlador basado en lógica difusa, que presenta una arquitectura orientada a cálculos paralelos, que es flexible a modificaciones inmediatas.

Todo el proceso difuso se puede ejecutar de manera asíncrona o bien, síncrona, en función del tipo de unidad divisora que se diseñe. En el caso de una unidad síncrona, se necesitan tantos ciclos de reloj como restas realice. La gran desventaja de la unidad de división mediante la segunda opción es que el hardware se incrementa en un 300% en comparación a la versión síncrona; sin embargo, es capaz de entregar el resultado de manera inmediata.

5. REFERENCIAS

- [1] Kandel Abraham, "FUZZY HARDWARE", Kluwer Academic Publishers, 1998.
- [2] Sebastian Michael J., "Application – Specific Integrated Circuits", Addison Wesley, 1998.
- [3] "Synopsis, FPGA Express with Verilog HDL and VHDL, Reference Manual", Xilinx in line, 1999.

Tecnología VPN (1ª parte)

**Lic. María Teresa Lozano Hernández,
Lic. María de Lourdes Olvera Cárdenas,
Ing. María del Rocío Velázquez Serrano,
Profesoras del CIDETEC-IPN**

Desde hace mucho tiempo, el mundo empresarial se enfrenta a la necesidad de implementar sistemas eficientes de manejo de información. La solución para muchas de estas empresas, independientemente de su actividad, ha sido las redes de telecomunicaciones, a través de las cuales han abatido costos y gastos de operación, manufactura, recursos humanos, etc. El uso de la tecnología de telecomunicaciones benefició a las empresas para agilizar sus ventas y básicamente sus movimientos contables, así como el servicio y soporte a los equipos sin tener que viajar entre ciudades o países. Las compañías prestadoras de servicios de telecomunicaciones comenzaron a invertir cada vez más en infraestructura para poder ofrecer mayor cantidad y calidad de servicios (QOS).

Esto también generó altos gastos de equipamiento para las empresas que se iniciaron en estas áreas, dudando muchas de ellas en invertir en equipo de comunicaciones. Por otro lado, la industria de las telecomunicaciones se vio favorecida y, hasta hace poco todavía, monopolizada. En la actualidad el beneficio de las telecomunicaciones se ha hecho cada vez mayor de tal forma que el número de servicios privados de los proveedores

se ha visto disminuido o le han dado un giro para aprovecharlo en otros servicios. Para las compañías clientes esto ha significado la reducción de hasta un 75% los gastos por renta de enlaces privados.

El porque de todo ello es el uso de las VPN's (Virtual Private Network) siendo estas la respuesta lógica al crecimiento de Internet; ya que permiten reemplazar líneas dedicadas por accesos a Internet con características adicionales. Se puede decir que una VPN es el uso de una infraestructura de telecomunicaciones pública para el uso privado o bien una conexión punto a punto de oficinas con usuarios remotos. Tal infraestructura pública está al alcance de casi cualquier persona o empresa a través de Internet. Sin embargo, por tratarse de una infraestructura pública es de esperarse que los datos pasaran a través de muchos puntos inseguros.

El objetivo principal de las VPN's es la seguridad y cualquier VPN que se precie de serlo deberá manejar los datos privados y confidenciales con un alto índice de seguridad. Una VPN típica cuenta con oficinas corporativas, una o más sucursales, cada una con sus enlaces a Internet de algún proveedor de servicios, un usuario remoto y un usuario móvil.

Aún cuando pareciera que una VPN es insegura, por el medio que utiliza, no lo es. Estas redes manejan un protocolo de seguridad para Inter-

net llamado IPSec y al mismo tiempo el protocolo estándar de IP, tanto para autenticar como para cifrar datos, de tal suerte que los equipos intermedios por donde pasa la información no distinguen entre un paquete IP y un paquete IPSec. Las VPN's también manejan certificados de seguridad como los bancos y otras instituciones. Los protocolos empleados tienen compatibilidad con equipos y aplicaciones. La seguridad se efectúa de extremo a extremo, ya que solo los equipos que se interconectan conocen el cifrado.

La red privada se extiende mediante un proceso de encapsulación y encriptación de los paquetes de datos a distintos puntos remotos mediante el uso de unas infraestructuras públicas de transporte. Los paquetes de datos de la red privada viajan por medio de un «túnel» definido en la red pública, ver **figura 1**. Conexión VPN

Las topologías pueden tener lógicas punto a punto, punto a multipunto; dependiendo de ello se generan las políticas de acceso así como también se forma el esquema de equipamiento de protección contra intrusos.

Antes de analizar las ventajas hablemos del caso en que se desea enlazar las oficinas centrales con alguna sucursal u oficina remota utilizando:

Modem: La desventaja es el costo de la llamada, ya que el costo de esta llamada sería por minuto conectado,

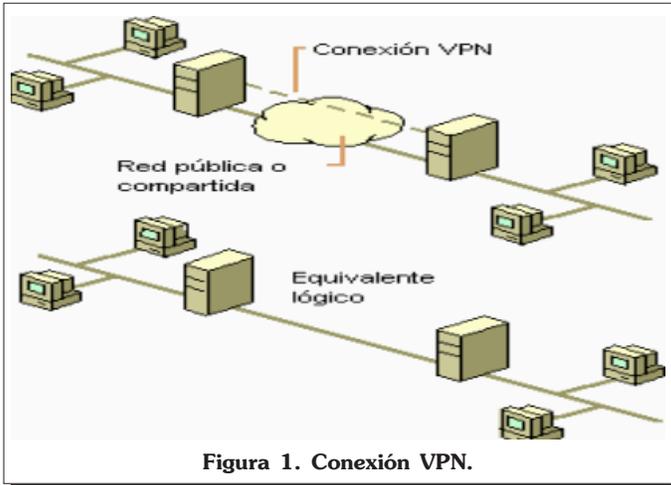


Figura 1. Conexión VPN.

además sería una llamada de larga distancia, a parte no se contaría con la calidad y velocidad adecuadas.

Línea Privada: Debe tenderse un cable ya sea de cobre o fibra óptica de un punto a otro, en esta opción el costo es muy elevado porque si se necesitara enlazar una oficina central con una sucursal que se encuentra a 200 Kilómetros de distancia el costo sería la renta mensual por Kilómetro, sin importar el uso.

VPN: Los costos son bajos porque solo se realizan llamadas locales, contando con que los datos viajan encriptados y seguros, con una buena calidad y velocidad, ver **figura 2**.

Los datos a través de una VPN pasan por el servidor dedicado, del cual parten, llegando a un firewall que a su vez hace la función de una pared para engañar a los intrusos, después los datos llegan a la nube de

los socios, con usuarios móviles, con oficinas remotas mediante los protocolos como Internet, IP, Ipsec, Frame Relay y/o ATM, como se muestra en la **figura 3**.

1 VENTAJAS DE UNA VPN

Hasta el momento solo podemos hablar de las bondades que nos ofrece el uso de las VPN's en una comparativa con las redes tradicionales. Dentro de las ventajas más significativas podremos mencionar la integridad, confidencialidad y seguridad de los datos. A continuación se enlistan algunas de estas:

- Reducción de costos.
- Sencilla de usar.
- Sencilla instalación del cliente en cualquier PC Windows.
- Manipulación en otros Sistemas Operativos
- Creación a partir de software o hardware
- Control de Acceso basado en políticas de la organización
- Herramientas de diagnóstico remoto.
- Manejo de direccionamiento propio
- Los algoritmos de compresión optimizan el tráfico del cliente.

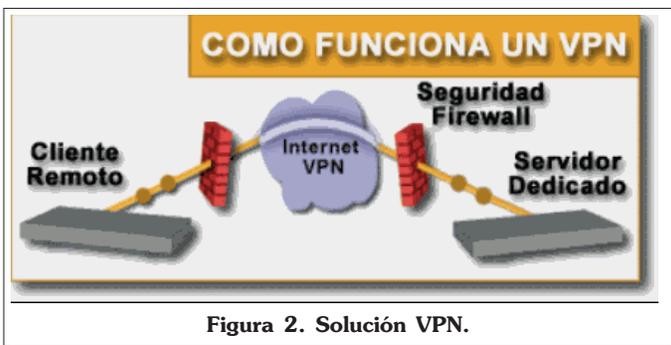


Figura 2. Solución VPN.

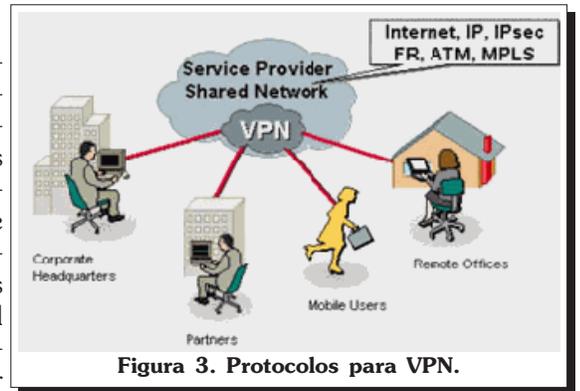


Figura 3. Protocolos para VPN.

- Evita el alto costo de las actualizaciones y mantenimiento a las PC's remotas.

Con respecto a la reducción de costo tenemos: reducción de costo de conectividad por líneas dedicadas eliminadas, menos accesos telefónicos, reducción de tiempo de configuración y mantenimiento ante una eventual caída del servidor o de la red misma, reducción de contratos de mantenimiento y configuración; eliminación de equipos PBX propiedad del cliente, sistemas de almacenamiento de energía o sistemas de energía ininterrumpible, sistemas de enfriamiento, etc., necesarios para el buen funcionamiento del cuarto de telecomunicaciones. Es importante recordar que el costo de los enlaces varía según la distancia.

2 OTRAS VENTAJAS

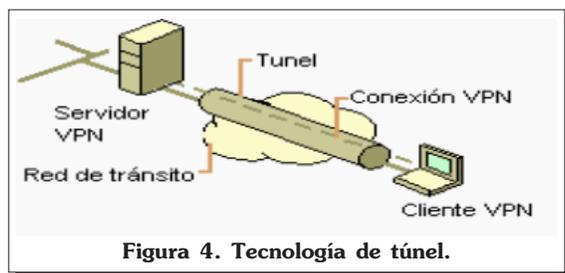
Incluyen la escalabilidad y flexibilidad de interconexión ya que permiten anexar nuevas conexiones o eliminar tantas conexiones como usuarios remotos se agreguen o dejen de utilizarla; la seguridad que involucra el cifrado de la información y la autenticación de los usuarios; diseño simplificado que reduce tiempos y costos de mantenimiento asociados a la gestión de la red; compatibilidad con protocolos de red más comunes (TCP/IP, IPX, NetBEUI), lo cual provoca el corrimiento de aplicaciones soportadas por estos protocolos, prioridad de tráfico ligado a las políticas de admi-

nistración de uso del ancho de banda a través de un administrador.

TECNOLOGÍA DE TÚNEL

Las redes privadas virtuales crean un túnel (**figura 4**) o conducto de un sitio a otro para transferir datos a esto se le conoce como encapsulación además los paquetes van encriptados de forma que los datos son ilegibles para los extraños.

El servidor busca mediante un ru-



teador la dirección IP del cliente VPN y en la red de tránsito se envían los datos sin problemas.

REQUERIMIENTOS BÁSICOS DE UNA VPN

Por lo general cuando se desea implantar una VPN hay que asegurarse que esta proporcione:

- Identificación de usuario
- Administración de direcciones
- Codificación de datos
- Administración de claves
- Soporte a protocolos múltiples

IDENTIFICACIÓN DE USUARIO

La VPN debe ser capaz de verificar la identidad de los usuarios y restringir el acceso a la VPN a aquellos usuarios que no estén autorizados. Así mismo, debe proporcionar registros estadísticos que muestren quien accede, que información y cuando.

2 ADMINISTRACIÓN DE DIRECCIONES

La VPN debe establecer una dirección del cliente en la red privada y debe cerciorarse que las direcciones privadas se conserven así.

2 CODIFICACIÓN DE DATOS

Los datos que se van a transmitir a través de la red pública deben ser previamente encriptados para que no puedan ser leídos por clientes no autorizados de la red.

2 ADMINISTRACIÓN DE CLAVES

La VPN debe generar y renovar las claves de codificación para el cliente y el servidor.

2 SOPORTE A PROTOCOLOS MÚLTIPLES

La VPN debe ser capaz de manejar los protocolos comunes que se utilizan en la red pública. Estos incluyen el protocolo de Internet (IP), el intercambio de paquete de Internet (IPX) entre otros.

Los dispositivos responsables para la formación y administración de la red virtual, para propiciar una comunicación con seguridad, deben ser capaces de garantizar:

La seguridad de los datos, en el caso que fuesen interceptados durante la transmisión, no pueden ser decodificados.

Integridad de los datos, además de no ser decodificados (seguridad), los datos no pueden ser modificados durante la transmisión.

La autenticación, garantía de que los datos están siendo transmitidos o recibidos del dispositivo remoto autorizado y no de un equipamiento cualquiera, o sea, garantía de que el dispositivo remoto con el cual fue establecido el túnel, es el dispositivo remoto autorizado y no otro equipamiento haciéndose pasar por él.

El servidor busca mediante un ruteador la dirección IP del cliente VPN y en la red de tránsito se envían los datos sin problemas.

HERRAMIENTAS DE UNA VPN

Las herramientas típicas de una VPN son:

- VPN Gateway
- Software
- Firewall
- Router

Los datos transitan codificados por Internet en «Túneles Virtuales» creados por los dispositivos VPN que utilizan criptografía; estos dispositivos son capaces de «entender» los datos codificados y forman una «red virtual» sobre la red pública. Es esa red virtual la que es conocida como VPN.

IP's PÚBLICAS, PRIVADAS Y SUS CLASES

Las IP's de todo el mundo son clasificadas en públicas y privadas. Las IP's públicas también conocidas como IP's homologadas, y se consiguen de manera permanente pagando una cantidad por la renta de cada IP que la compañía solicita. Éstas IP's tienen la característica de ser globalmente ruteables, mientras que las IP's no homologadas solo pueden ser ruteadas en un ambiente totalmente privado.

La división para las IP's versión 4 se lleva a cabo por una división en clases.

Formato	Bits de orden superior	Rango direcciones de	Número de bits de red y host	Máximo de hosts
RH.H.H	1/0	1.0.0.0 a	7/24	2 ²⁴ -2
RRH.H	2/10	128.1.0.0 a 191.254.0.0	14/16	2 ¹⁶ -2
RRRH	3/110	192.0.1.0 a 223.255.254.0	22/8	2 ⁸ -2
No utilizado	4/1110	224.0.0.0 a 239.255.255.255	No se utilizan	No utilizados
No utilizados	4/1111	240.0.0.0 a 254.255.255.255	No se utilizan	No utilizados

Tabla 1. Clases de redes IP's.

Rango de direcciones	Clase	Número de redes
10.0.0.0 10.255.255.255	A	1
172.16.0.0 172.31.255.255	B	16
192.168.0.0 192.168.255.255	C	256

Tabla 2. Rango para direcciones privadas.

Las clases de redes IP's se muestran en la **tabla 1** y el rango para direcciones privadas se muestra en la **tabla 2**.

INFRAESTRUCTURA PARA UNA VPN

EQUIPAMIENTO

Para poder crear una VPN debemos tener acceso a Internet mediante cualquier tecnología disponible en nuestro centro de trabajo ya sea inalámbrica, mediante POTS, ISDN, ADSL o algún otro medio de acceso hacia la red. Es de hacer notar que la velocidad de acceso irá estrechamente relacionada con la cantidad de información que se desee pasar por el túnel de la VPN, es obvio que las oficinas centrales deberán contar con un enlace de mayor capacidad ya que estas proveerán la mayor parte de la información para ser accedidas por los usuarios y las oficinas remotas.

Teniendo el enlace se requiere una solución para manejar la(s) VPN(s), esto puede llevarse a cabo mediante software o hardware, las soluciones por hardware son más recomendables, ya que son las que tienen un mejor performance, además permiten un mejor rendimiento del

enlace que utilice para acceder a servicios públicos y a las VPN's.

SEGURIDAD EN LOS ENLACES

REDUNDANCIA

Para una VPN que corre aplicaciones críticas es necesario manejar algún tipo de redundancia, existen proveedores de equipo que manejan una redundancia en sus equipos al hacer que soporten un sistema de fail-over.

Un sistema de fail-over consiste en tener un equipo con redundancia en la comunicación WAN, de tal suerte que se pueda detectar cuando falla el enlace principal y de esta manera conmutar la comunicación al enlace de respaldo, el enlace principal seguirá siendo censado para detectar cuando vuelva a estar arriba para volver a éste.

ENCRIPTAMIENTO

Las redes VPN basadas en IPsec son de mayor demanda en estos días, por lo tanto nos enfocaremos a describir el estándar de IPsec.

PROTOSCOLOS DE VPN (IPSEC)

Las redes se diseñan normalmente para impedir el acceso no autorizado a datos confidenciales desde fuera

de la intranet de la empresa mediante el cifrado de la información que viaja a través de líneas de comunicación pública. Sin embargo, la mayor parte de las redes manejan las comunicaciones entre los hosts de la red interna como texto sin formato. Con acceso físico a la red y un analizador de protocolos, un usuario no autorizado puede obtener datos privados.

IPsec autentifica los equipos y cifra los datos para su transmisión entre hosts en una red, intranet o extranet, incluidas la comunicaciones entre estaciones de trabajo y servidores, y entre servidores. El objetivo principal de IPsec es proporcionar protección a los paquetes IP. IPsec esta basado en un modelo de seguridad de extremo a extremo, lo que significa que los únicos hosts que tienen que conocer la protección de IPsec son el que envía y el que recibe. Cada equipo controla la seguridad por sí mismo en su extremo, bajo la hipótesis de que el medio por el que se establece la comunicación no es seguro.

IPsec aumenta la seguridad de los datos de la red mediante:

- La autenticación mutua de los equipos antes del intercambio de datos. IPsec puede utilizar kerberos V5 para la autenticación de los usuarios.
- El establecimiento de una asociación de seguridad entre los dos equipos. IPsec se puede implementar para proteger las comunicaciones entre usuarios remotos y redes, entre redes e incluso, entre equipos cliente dentro de una red de área local (LAN).
- El cifrado de los datos intercambiados mediante **Cifrado de datos estándar** (DES, Data Encryption Standard), triple DES (3DES) o DES de bits. IPsec usa formato de paquete IP estándar en la autenticación o el cifrado de los datos. Por tanto, los dispositivos de

red intermedios, como los enrutadores, no pueden distinguir los paquetes de IPsec de los paquetes IP normales.

El protocolo también proporciona las siguientes ventajas:

- Compatibilidad con la infraestructura de claves públicas. También acepta el uso de certificados de claves públicas para la autenticación.
- Compatibilidad con claves compartidas. Si la autenticación mediante kerberos V5 o certificados de claves públicas no es posible, se puede configurar una clave compartida (una contraseña secreta compartida) para proporcionar autenticación y confianza entre equipos.
- Transparencia de IPsec para los usuarios y las aplicaciones. Como IPsec opera al nivel de red, los usuarios y las aplicaciones no interactúan con IPsec.
- Administración centralizada y flexible de directivas mediante directiva de grupo. Cuando cada equipo inicia una sesión en el dominio, el equipo recibe automáticamente su directiva de seguridad, lo que evita tener que configurar cada equipo individualmente. Sin embargo, si un equipo tiene requisitos exclusivos o es independiente, se puede asignar una directiva de forma local.
- Estándar abierto del sector. IPsec proporciona una alternativa de estándar industrial abierto ante las tecnologías de cifrado IP patentadas. Los administradores aprovechan la interoperabilidad resultante.

ATAQUES A LA SEGURIDAD

A continuación se presenta una lista parcial de los ataques más comunes a las redes tradicionales:

- Rastreo. Un rastreo de red es una aplicación o un dispositivo que puede supervisar y leer los paquetes de la red. Si los paquetes no están cifrados, un rastreo de red obtiene una vista completa de los datos del paquete. El monitor de red de Microsoft es un ejemplo de rastreador de red.
- Modificación de datos. Un atacante podría modificar un mensaje en tránsito y enviar datos falsos, que podrían impedir al destinatario recibir la información correcta o permitir al atacante conseguir la información protegida.
- Contraseñas. El atacante podría usar una contraseña o clave robada, o intentar averiguar la contraseña si es fácil.
- Suplantación de direcciones. Este ataque va dirigido a servidores de aplicaciones al explotar las debilidades del sistema operativo y de las aplicaciones del servidor.
- Intermediario. En este tipo de ataque, alguien entre los dos equipos comunicantes está supervisando activamente, capturando y controlando los datos de forma desapercibida, (el atacante puede estar cambiando el encaminamiento de un intercambio de datos).
- Denegación de servicio. El objetivo de este ataque es impedir el uso normal de equipos o recursos de la red. Por ejemplo, cuando las cuentas de correo electrónico se ven desbordadas con mensajes no solicitados.

IPSEC

CARACTERÍSTICAS DE SEGURIDAD DE IPSEC

Las siguientes características de IPsec afrontan todos estos métodos de ataque:

- Protocolo Carga de seguridad de encapsulación (ESP, Encapsulating Security Payload). ESP proporciona privacidad a los datos mediante el cifrado de los paquetes IP.
- Claves basadas en criptografía. Las claves cifradas, que se comparten entre los sistemas que se comunican, crean una suma de comprobación digital para cada paquete IP. Cualquier modificación del paquete altera la suma de comprobación, mostrando al destinatario que el paquete ha sido cambiado en su tránsito. Se utiliza material de claves diferentes para cada segmento del esquema de protección global y se puede generar un nuevo material de claves con la frecuencia especificada en la directiva de IPsec.
- Administración automática de claves. Las claves largas y el cambio dinámico de claves durante las comunicaciones ya establecidas protegen contra los ataques. IPsec usa el protocolo Asociación de seguridad en internet y administración de claves (ISAKMP, Internet Security Association and Key Management Protocol) para intercambiar y administrar claves cifradas entre los equipos que se comunican.
- Negociación de seguridad automática. IPsec usa ISAKMP para negociar de forma dinámica un conjunto de requisitos de seguridad mutuos entre los equipos que se comunican. No es necesario que los equipos tengan directivas idénticas, solo una directiva configurada con las opciones de negociación necesarias para establecer un conjunto de requisitos con otro equipo.
- Seguridad a nivel de red. IPsec existe en el nivel de red, proporcionando seguridad automática a todas las aplicaciones.
- Autenticación mutua. IPsec permite el intercambio y la comprobación de identidades sin exponer

la información a la interpretación de un atacante. La comprobación mutua (autenticación) se utiliza para establecer la confianza entre los sistemas que se comunican. Sólo los sistemas de confianza se pueden comunicar entre sí. Los usuarios no tienen que estar en el mismo dominio para comunicarse con la protección de IPsec. Pueden estar en cualquier dominio de confianza de la empresa. La comunicación se cifra, lo que dificulta la identificación e interpretación de la información.

Filtrado de paquetes IP. Este proceso de filtrado habilita, permite o bloquea las comunicaciones según sea necesario mediante la especificación de intervalos de direcciones, protocolos o, incluso, puertos de protocolo específicos.

COMPONENTES DE IPSEC

En el proceso de autenticación y cifrado de IPsec intervienen varios componentes. Su conocimiento y el de los procesos en que consiste la comunicación IPsec le ayudará a encontrar soluciones a los problemas de implementación.

EL PROCESO DE NEGOCIACIÓN Y FILTRADO

Cuando un equipo configurado con una directiva de IPsec intenta comunicarse con otro equipo, comienza el proceso siguiente:

1. Las directivas de IPsec se entregan al controlador de IPsec y el intercambio de clave ISAKMP/Oakley a través de directivas locales o configuraciones de directivas de grupo.
2. ISAKMP supervisa las negociaciones entre los hosts y proporciona claves que se usan con algoritmos de seguridad.
3. El controlador de IPsec supervisa, filtra y protege el tráfico entre el nivel de transporte y el nivel de red.

CONCLUSIÓN

Las VPN representan una gran solución para las empresas en cuanto a seguridad, confidencialidad e integridad de los datos y prácticamente se ha vuelto un tema importante en las organizaciones, debido a que reduce

significativamente el costo de la transferencia de datos de un lugar a otro, así como el de omitir gastos en telefonía local y de larga distancia y otros servicios, el único inconveniente que pudieran tener las VPN es que primero se deben establecer correctamente las políticas de seguridad y de acceso, porque si esto no está bien definido pueden existir consecuencias serias de interferencia por intrusos. En la siguiente publicación abordaremos otros temas relacionados con la puesta en pie de una VPN.

BIBLIOGRAFÍA

- [1] Uyles Black, *"Tecnologías Emergentes de Telecomunicaciones"*, 2a. edic. Prentice Hall.
- [2] Williams Stallings, *"Handbook of Computer – Communications Standards Vol. 3"*, Stallings / Macmillan Books.
- [3] Stephen J. Bigelow, *"Herramientas, mantenimiento y reparación de Redes"*, Mc Graw Hill

Workflow: Tecnología Para la Innovación Organizacional

Lic. Elizabeth Acosta Gonzaga
Profesora del CIDETEC-IPN
M. en C. Abraham Gordillo Mejía
Profesor de UPIICSA-IPN

La búsqueda de mayor productividad en las organizaciones es un objetivo básico de trabajo para conseguir más eficiencia y competitividad en el mercado; actualmente existen varias formas de lograrlo, desde estudios de reducción de costos, hasta la mejora de procesos mediante la automatización y reordenamiento de actividades, etc. De esta forma tanto la administración pública como la privada deben repensar todos sus procesos de una forma sinérgica; Ya sea que lo llamen transformación de los negocios o reestructuración de los procesos de negocio. Esto significa que las organizaciones deben rediseñar los procesos críticos e invertir en tecnología.

En el momento en que las organizaciones rediseñan los procesos deben continuar para auto modificarse, para reflejar los cambios que se producen en el ambiente en que se desenvuelven, y responder rápida y eficientemente tanto a clientes como a proveedores.

El propósito de este artículo es presentar una herramienta denominada "Workflow" y el procesamiento de imágenes como dos tecnologías de información críticas para apoyar la reestructuración de los procesos de

negocio. Para ello es importante definir que es un proceso de negocio.

PROCESO DE NEGOCIO

Un proceso de negocio es un grupo estructurado de actividades (procedimiento administrativo) que se ejecutan por personas, aplicaciones o sistemas de cómputo. Cualquier actividad de negocio puede ser clasificada como un proceso: sus procedimientos actuales de compra o venta, lanzar o discontinuar productos, invertir o controlar gastos, cobranzas o pagos, son todos ejemplos de procesos de negocio. Cualquier organización, ya sea de manufactura, distribución, servicio o gobierno es una red de procesos entrelazados interna y externamente.

REESTRUCTURACIÓN DE LOS PROCESOS DE NEGOCIO

En términos simples, la reestructuración de los procesos de negocio es el rediseño de los mismos mediante tecnologías de información. El objetivo es alcanzar mejoras importantes en calidad, productividad y servicio. De este modo, la tecnología se usa no solamente para automatizar o alterar procesos existentes, sino también como herramienta para cambiar fundamentalmente la forma en que las organizaciones realizan sus actividades que hacen, es decir, **innovar**.

¿QUÉ SON LOS WORKFLOW?

Los Workflows son sistemas que ayudan a administrar y automatizar procesos de negocios; un Workflow puede ser descrito como el flujo y control en un proceso de negocio.

La WfMC (Workflow Management Coalition) define a los Workflows como:

«La automatización de un proceso de negocio, total o parcial, en la cual documentos, información o tareas son transferidos o encadenados (as) de un empleado a otro, conforme lo estipule un procedimiento de trabajo». Cabe resaltar que los Workflows son sólo una alternativa para el manejo de la información, para reducir tiempo, dinero y esfuerzo en la ejecución de un proceso de negocio.

Por definición, el **software para Workflow** provee un medio de automatizar y controlar completamente el trabajo, tal y como este definido, asignado, priorizado y distribuido entre la gente de un mismo grupo, en áreas separadas o a todo lo largo de la organización, y que esté plasmado en un Manual de Organización o de Procedimientos.

Idealmente, debe incluir un conjunto de herramientas gráficas para definir procesos de negocios propios sin la necesidad de aprender un lenguaje de programación.

El software de *Workflow* supone una situación especial en las organizaciones a fin de explotar al máximo sus potencialidades de modelaje y proceso; esta situación consiste en que los procesos de trabajo estén en constante cambio, es decir, las condiciones del negocio se modifiquen constantemente como respuesta a un mercado dinámico. Por ejemplo, las siguientes situaciones deberían estar presentes: la gente es reubicada, aparecen nuevas fuentes de información o se encuentran nuevos proveedores que condicionan formas de administrar, o bien se contratan más maquileros y/o distribuidores, etc., de tal forma que se puedan usar herramientas gráficas para modificar los procesos en el momento que se requiera. Al mejorar continuamente los patrones de trabajo y refinar la productividad las organizaciones se logran las metas de reestructuración de los procesos de negocio, para mejorar servicios, calidad y competitividad.

SISTEMAS DE IMÁGENES QUE AYUDAN A LOGRAR LOS OBJETIVOS

Workflow y sistema de imágenes no son sinónimos. El *Workflow* ayuda a lograr un aumento significativo en la productividad y eficiencia de sus procesos de negocio; para lograrlo los programas de *Workflow*, en muchos casos, tienen capacidades de manejo de imágenes asociadas a ellos.

Es cierto que los programas para *Workflow* pueden ser incorporados a aplicaciones tradicionales sin capacidades de manejo de imágenes, pero para que las organizaciones obtengan el beneficio total del *Workflow*, el procesamiento de imágenes es la herramienta que provee control y flexibilidad; sin procesamiento de imágenes el software *Workflow* automatiza un número menor de procesos.

La **figura 1** muestra un ejemplo de un documento y un flujo por donde este debe pasar, dando la posibilidad de redistribuir su curso en las diferentes áreas o puestos.

BENEFICIOS DEL SISTEMA WORKFLOW

Uno de los problemas comunes en el desarrollo de aplicaciones es que los procesos que se desarrollan en el entorno laboral quedan en el código de la aplicación que resuelve el problema, y la mayoría de los usuarios no tiene el conocimiento necesario para realizar cambios en dichos procesos.

Una solución apropiada a este problema es separar los procedimientos y asociarlos a los flujos de trabajo dentro de la empresa; como vemos, el *Workflow* se relaciona con la automatización de los procedimientos donde los documentos, la información o tareas pasan entre los participantes del sistema de acuerdo a un conjunto de reglas previamente establecidas con la finalidad de llegar a culminar una meta común.

Otra ventaja de éstos sistemas es que pueden eficientar un proceso de trabajo a fin de reducir el tiempo que se pierde para acceder a documentos necesarios, bien para su llenado, o por su pérdida, entre otras situaciones, por lo que los *Workflow* proponen impactar el manejo de estos documentos mediante el manejo y proceso de imágenes de (manejo de formularios u otra forma electrónica), lo que redundará en una disminución significativa en la cantidad de papeles manejados.



Figura 1. Ejemplo de imágenes de un software *Workflow*.

Una vez capturado e indexado un procedimiento o proceso en un sistema de imágenes digitalizadas es posible integrar y compartir todos sus ventajas en una red de cómputo. Integrando programas de *Workflow* con procesamiento de imágenes se puede controlar la manera en que las imágenes pueden ser integradas o combinadas con otros tipos de información o aplicaciones, y ser distribuidas los empleados que lo requieran.

MEJOR ACCESO A LA INFORMACIÓN

El procesamiento de imágenes facilita el acceso a archivos electrónicos eliminando la necesidad de esperar que los expedientes o archivos basados en papel pasen de una persona a otra. Distintos empleados en distintas computadoras pueden completar su parte del proceso simultáneamente. Esta transferencia electrónica y el compartir información acelera significativamente los procesos que requieren varias etapas y múltiples fuentes de información.

Las funciones más comunes que proporcionan los *Workflows* son:

- Asignación de tareas al personal.
- Aviso al personal de tareas pendientes.
- Permitir la colaboración en las tareas comunes.
- Optimización de recursos humanos y técnicos, alineándolos a la estrategia de la empresa.
- Automatización y optimización de las secuencias de los procesos de negocio.
- Agilización de los procesos de negocio, proporcionado así un mejor servicio al cliente.
- Control y seguimiento estricto de dichos procesos.

DIRECCIONAMIENTO INTELIGENTE

Algunos procesos requieren un direccionamiento secuencial automatizado: las tareas de un proceso son direccionadas electrónicamente de una persona a otra, es decir, de una ubicación a otra. En un departamento de cuentas por pagar, por ejemplo, una factura puede ser procesada en la computadora de una persona, colocada en espera hasta que la información adicional que se requiera llegue, y luego automáticamente transferida a la computadora de la persona que realiza el trabajo final.

Otros procesos requieren procesamiento concurrente o paralelo. El trabajo puede ser direccionado concurrentemente a través de varios departamentos para completar enteramente las funciones requeridas. Los programas de *Workflow* deben proveer la capacidad de esperar por la terminación de todas las tareas necesarias, colocar los resultados todos juntos y presentarlos a la persona siguiente en el proceso. Por ejemplo, una solicitud de crédito puede ser direccionada al departamento de créditos para revisar el estado financiero del solicitante, y al departamento de ventas para autorizar un nuevo crédito u

otra venta. Ambos reciben la solicitud de crédito concurrentemente y pueden completar su trabajo simultáneamente. El sistema espera por ambos, enviando los correspondientes recordatorios o avisos si el tiempo establecido ha expirado, y luego envía la solicitud con los informes a la oficina de crédito responsable por la aprobación final.

Hay procesos que requieren direccionamiento condicional, en el cual el procesamiento del trabajo es determinado por una aplicación tradicional o la acción de un empleado o una información identificada dentro del trabajo. Por ejemplo, si el reclamo por un seguro de auto es de más de \$10.000 pesos, este será direccionado y procesado en forma diferente que uno por menor valor.

Los programas de *Workflow* deben proveer direccionamiento inteligente junto con la habilidad para manejar procesos excepcionales o manejos especiales, un ejemplo puede ser la necesidad de intervención gerencial o la aprobación previa antes de que la tarea pase al próximo puesto.

¿CUÁNDO UTILIZAR UN WORKFLOW?

Existe sin duda una justificante "ideal", que consiste en proyectos institucionales o de alta gerencia que pretenden transformar la organización, ya sea en un modelo inteligente rediseñando sus procesos actuales de trabajo en la búsqueda del aumento de la productividad y la calidad, o en mejorar la imagen empresarial, entre otros. Sin embargo, existirán negocios que buscarán simplemente introducir algunos elementos de innovación tecnológica sumados a la presencia en la organización de redes locales, intrainstitucionales o extrainstitucionales.

Para cualquiera de estos proyectos, el siguiente cuestionario puede ayudar a identificar posibles oportunidades con alto potencial para desarrollar proyectos de imágenes y de *Workflow*.

- ¿Hay una cantidad significativa de documentos y papeles involucrados en los procedimientos?
- ¿Existen muchos puestos asociados al manejo de documentos?
- ¿Los documentos usados, están asociados a los procesos de toma de decisiones?
- ¿Hay una pesada carga de trabajo generada por archivo -recuperación - archivo?
- ¿Más de un usuario necesita el mismo documento?
- ¿Lo necesitan al mismo tiempo?
- ¿Se hacen múltiples fotocopias regularmente?
- ¿Se necesitan los documentos en más de un lugar?
- ¿Se pierden frecuentemente los documentos?
- ¿El trabajo es impactado significativamente por el manejo de documentos?

Si la respuesta a varias de estas preguntas es afirmativa, entonces es recomendable un sistema *Workflow* y de imágenes.

Entonces, es necesario evaluar a un nivel más analítico, planteándose las siguientes preguntas:

- ¿Qué rutas siguen los documentos (mapeo de procesos)?
- ¿Qué puestos participan en cada proceso?
- ¿Cuál es el rol que juega cada participante?
- ¿Qué decisiones son tomadas?
- ¿Cómo se llevan a cabo estas decisiones?
- ¿Qué información es requerida por cada participante?

Estas preguntas son indispensables para poder identificar correctamente los procesos de negocio que pueden ser mejorados e implementados a través de un *Workflow*.

TRES TIPOS DE WORKFLOW

Workflow se define típicamente en tres categorías, que van desde las más complejas y estructuradas hasta las que usan correo electrónico.

DE TRANSACCIONES

El *Workflow* orientado a transacciones es usado en aplicaciones tradicionales gobernadas por una serie de normas y procedimientos. Requieren personal para realizar tareas repetitivas en las cuales los documentos pueden requerir ser accedidos por pedido (días, meses o aun años después). Además, se requieren reglas para crear y mantener un registro de auditoría de cada documento. Ejemplos de este tipo incluyen líneas de crédito, reclamos, etc.

DE COLABORACIÓN

Las aplicaciones de *Workflow* que resuelven procesos de negocio (business process) donde participa gente para lograr una meta común se conocen como *Workflow* de Colaboración; éstos estructuran o semi-estructuran procesos de negocio donde participa gente con el objetivo de lograr una meta en común.

Típicamente involucran documentos los cuales son los contenedores de la información, se sigue la ruta de estos paso a paso además de las acciones que se toman sobre ellos.

Los documentos son la clave; es esencial para la solución de *Workflow* mantener la integridad de los documentos.

A LA MEDIDA

El *Workflow* orientado a proyectos o Ad-hoc incluye un grupo indefinido de personas con fechas específicas para realizar tareas. Este tipo de *Workflow* implica una gran cantidad de tiempo para su coordinación. Es típicamente de corta vida y no estructurado, variando mucho en su complejidad. Ejemplos de este tipo son: desarrollo de planes estratégicos, diseño de productos, evaluación de un producto, etc.

- Tareas.
- Personas (Usuarios).
- Roles.
- Rutas.
- Reglas de Transición.
- Datos.
- Eventos.
- Plazos (Deadlines).
- Procesos.
- Políticas.

CONCLUSIÓN

Algunas de las soluciones actuales que incluyen herramientas *Workflow* ayudan a las organizaciones a mejorar el flujo del trabajo, acelerando los servicios, mejorando la calidad, la productividad y competitividad. Combinado con los beneficios del procesamiento de imágenes, estas soluciones proveen un medio versátil para optimizar el trabajo tanto a nivel de la organización como de los departamentos.

Con estas soluciones, las organizaciones son capaces de hacer más y mejor lo que hacen; mejorando sus operaciones según sean sus necesidades para no quedar atrapadas en los procesos existentes.

MODELANDO WORKFLOW

A pesar de la gran variedad de alternativas que se encuentran en el Mercado de éstos sistemas, se puede ver que los conceptos utilizados que no varían en gran forma; esto permite que se tienda a realizar un modelo de implementación general.

La tendencia actualmente es identificar los principales componentes de un sistema de *Workflow* para representarlos dentro de un mismo modelo abstracto.

Es necesaria la representación formal de un modelo que permita la realización de sistemas sobre diversos escenarios, posibilitando que distintos sistemas de *Workflow* puedan interactuar entre sí.

CONCEPTOS INCLUIDOS PARA MODELAR EL WORKFLOW

Cuando se modela un sistema de *Workflow* generalmente se identifican y utilizan definiciones de los distintos elementos que se pueden encontrar dentro de dicho sistema. A continuación listamos estos elementos, para luego dar una descripción o definición de cada uno de ellos:

BIBLIOGRAFÍA

- [Khoshafian]** S. Khoshafian. *"Introduction to Groupware, Workflow and WorkGroup Computing"*. Editorial Wiley, 1995. ISBN: 0-471-02946-7.
- [Coleman97]** David Coleman. *"Groupware Collaborative Strategies for Corporate and Intranets"*. Editorial Prentice Hall, 1997. ISBN: 0-13-727728-8
- [Raymond]** Raymond L. Manganelli, Mark M. Klein. *Como Hacer Reingeniería*. Grupo Editorial Norma. ISBN: 958-0430-25-x.
- [Booch97]** Martin Fowler, Kendall Scott. *UML Distilled*. Addison Wesley Longman, Inc., 1997. ISBN: 0-201-32563-2
- [WFMC]** *"Workflow Management Coalition. The Workflow Reference Model"*. <http://www.aiai.ed.ac.uk/project/wfmc>.
- [WFSteffen]** Dr Hermann Steffen. *"Workflow. Una Herramienta para la Gestión de Procesos de Empresa"*.
- [KenOrr]** Ken Orr Institute. *"Visual Enterprise Modeler"*.
- [Ensemble]** FileNet Corporation. *"FileNet Ensemble User's Guide"*.
- [ODMVC]** StingraySoftware. *"The Model-View-Controller Framework"*.
<http://www.stingray.com>
- [ODGuide]** Stingray Software. *"Guide to Using Objective View"*.
<http://www.stingray.com>
- [UMLEBM]** UML. *"UML Extension for Business Modeling"*.
<http://www.rational.com/uml>
- [UMLNGuide]** UML. *"UML Notation Guide"*.
<http://www.rational.com/uml>
- [WFGW]** Ultimus. *"Workflow, Groupware and the rol of Ultimus"*.
<http://www.ultimus1.com>

PCI Express: Una Visión Inicial

**M. en C. Eduardo Rodríguez Escobar,
Lic. Elizabeth Acosta Gonzaga**
Profesores del CIDETEC-IPN.

Luando se hace referencia al rendimiento de un equipo de cómputo (especialmente en las computadoras personales, PC), el principal aspecto considerado es la velocidad del procesador. Sin embargo, cuando hablamos de computadoras de propósito general, la velocidad de transferencia de datos puede ser un parámetro tan o incluso más importante en un momento dado. Comparado con el avance continuo en diversas áreas de la tecnología de cómputo, tal como los procesadores o las tarjetas de video, el adelanto en la tecnología de los sistemas de entrada/salida (E/S) parece estar detenido.

Desde la aparición del bus PCI (*Peripheral Component Interconnect*) en su versión original en junio de 1992, realmente ha cambiado poco la manera en que los datos se manejan dentro de la computadora. Sin embargo, probablemente sea bueno que el bus PCI haya permanecido, en esencia, sin cambios desde su aparición; como quiera que sea trabaja. Por casi una década, este bus ha proporcionado una plataforma estable y flexible para los desarrolladores de hardware y software. Cualquiera que recuerde los días anteriores a

Windows 95 y a los dispositivos *Plug and Play*, puede entender porque el uso de las computadoras es más común desde entonces.

Actualmente, tanto los discos duros como los dispositivos periféricos (tarjetas de red, sonido, etc.) y aun los puertos USB, transfieren los datos a través de un sistema de entrada/salida similar al de la primera PC con procesador Intel 80486; es decir, el bus PCI corriendo a 32 bits/33MHz, con tasas de transferencia de 133MB/s. Esto evidentemente es un cuello de botella, si se compara con la velocidad de transferencia entre otros subsistemas; por ejemplo, una computadora personal con microprocesador Pentium 4 y memoria DDR (ciertamente un sistema básico en la actualidad) puede intercambiar hasta 2.1 GB/s de datos a través del bus de memoria, mientras que el bus AGP 8x puede transferir información también en este mismo rango⁽¹⁾.

Un disco duro ATA 133 conectado a una controladora IDE UltraATA/133 puede, teóricamente, sobrecargar al bus PCI, al menos durante periodos pequeños. Aún así, la tasa de transferencia sostenida de los discos duros actuales es demasiado baja para ocasionar problemas.

Otros dispositivos tales como los arreglos RAID, los ya venerables SCSI y las tarjetas Gigabit Ethernet, por mencionar algunos, también intentan enviar información a través del bus PCI paralelo con su limitada capacidad de 133MB/s, lo cual representa claramente una situación de congestión.

Hoy en día existen cuatro especificaciones adicionales de PCI, todas diseñadas para incrementar el ancho de banda disponible; éstas se indican en la **Tabla 1**.

El problema es que, si bien estas tecnologías tienen o tendrán pronto un lugar permanente en el mercado de servidores, la complejidad y el costo extra que representan para la fabricación de la tarjetas madre significa que serán virtualmente desconocidas al nivel de computadoras de escritorio. El PCI-X, por ejemplo, requiere un controlador para cada ranura, lo que lo hace muy costoso; sin

Tipo de Bus PCI	Ancho (bits)	Velocidad (MHz)	Ancho de Banda (MB/s)
PCI	32	33	133
PCI 66MHz	32	66	266
PCI 64-bit	64	33	266
PCI 66MHz/64 bit	64	66	533
PCI-X	64	133	1,066

Tabla 1. Tipos de Bus PCI

⁽¹⁾ Entre paréntesis, los autores no consideramos al AGP un verdadero bus de conexión a periféricos (si, en plural), al ser prácticamente exclusivo de las tarjetas de video y contar los sistemas únicamente con una ranura o *slot* de conexión de este tipo.

embargo, la solución a esto ya está en proceso y cuenta con el respaldo del fabricante de procesadores favorito de las mayorías, Intel.

Intel, en cooperación con varias compañías tales como: IBM, Dell, HP y Microsoft, introdujo lo que se espera sea el nuevo estándar para la E/S de los equipos personales en los próximos años: el PCI Express (también denominado **3GIO** o Tercera Generación de E/S). Este nombre ha sido acuñado por el PCI-SIG (*PCI Special Interest Group*), que es el comité responsable de vigilar el estándar de la interfase PCI. Esta norma, aprobada el 17 de abril del 2002, mantiene una compatibilidad completa en hardware y software con todos los dispositivos PCI recientes. Sin embargo, en términos de formato es algo completamente nuevo.

TECNOLOGÍAS DE BUS Y EL PCI

Antes de analizar al PCI Express (lo cual será tema de un artículo posterior), es importante entender que tanto ha avanzado la tecnología de los buses. En 1984, IBM lanzó su PC AT, donde el procesador, la memoria y el bus de E/S compartían un reloj común de 8 MHz. Este bus se conoció como el estándar ISA (*Industry Standard Architecture*), el cual tiene una interfase de 16 bits, lo que significa que los datos únicamente pueden transferirse dos bytes a la vez; más aún, normalmente requiere de dos o tres señales de reloj para transferir esos dos bytes de datos. Esto no era un problema para los dispositivos que son inherentemente lentos (puertos COM, puertos de impresora, tarjetas de sonido o unidades de CD-ROM); sin embargo, el bus ISA era muy lento para los accesos a disco de alto desempeño y las tarjetas de video con mayor resolución. Esto provocó el surgimiento de otras arquitecturas,

tales como el microcanal (MCA) y los buses EISA y VESA local (VLB), que tampoco constituyeron una solución lo suficientemente rápida. Finalmente, fue el bus PCI el que pudo aumentar el ancho de banda del sistema.

EL PCI es una interfase de 64 bits en un empaque de 32 bits; imaginar esto requiere un poco de aritmética. El bus trabaja a 33 MHz, y puede transferir de 32 bits de datos (4 bytes) en cada ciclo de reloj, lo cual suena como un bus de 32 bits. Sin embargo, la duración de un pulso de reloj a 33 MHz es de 30 nanosegundos, mientras que la velocidad de la memoria es de 70 nanosegundos. Cuando el CPU toma datos de la RAM, tiene que esperar al menos 3 pulsos de reloj por dichos datos. Al transferir datos en cada pulso de reloj, el bus PCI puede proporcionar el mismo rendimiento con una interfase de 32 bits que el de otras partes de la máquina con un grupo de 64 bits.

SOLUCIONES PROPUESTAS

PCI-X

Aunque en un principio el PCI representó un gran avance, la aparición de discos duros y periféricos PCI más rápidos ha provocado que este bus se congestione tanto como la Ciudad de México en las horas pico. Esto hizo evidente la necesidad de un nuevo estándar que proporcionara un mayor ancho de banda. PCI-X es un paso intermedio entre PCI y PCI Express, el cual fracasó en equipos de

escritorio, aunque el PCI Consortium insiste en que no va a desaparecer. PCI-X emplea la misma interfase paralela que PCI, ofreciendo un ancho de banda superior a algunas implementaciones de PCI Express (hasta 4.3GB/seg para PCI-X 533). PCI-X tiene una ventaja: es compatible con las tarjetas PCI actuales, mientras que PCI Express necesita hardware nuevo, razón por la cual PCI-X se mantendría por un tiempo en el mercado de servidores.

PCI-EXPRESS

PCI Express es un desarrollo para aumentar la funcionalidad de E/S en las computadoras actuales. Este diseño tuvo como finalidad recuperar el balance entre las velocidades del CPU y el sistema, considerando la demanda siempre creciente de procesamiento rápido y que el CPU se encontraba bloqueado por los buses de sistema obsoletos. En el diagrama siguiente (**Figura 1**) se muestra el considerable aumento en el ancho de banda del PCI Express.

PCI Express está diseñado no sólo para sustituir al bus PCI para dispositivos periféricos tales como modems y tarjetas de red, sino también al bus AGP para tarjetas gráficas. Al contra-

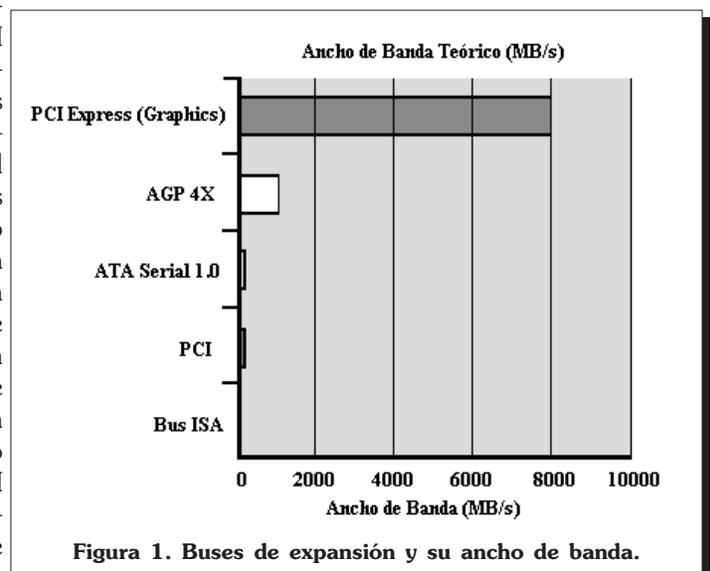


Figura 1. Buses de expansión y su ancho de banda.

rio que su predecesor paralelo, PCI Express es un sistema de interconexión serie punto a punto, capaz de ofrecer una alta tasa de transferencia con un alto ancho de banda, desde 250MB/seg para la implementación 1X, hasta 4GB/seg para el PCI Express 16X que se emplea en las tarjetas gráficas; la notación 1X y 16X se refiere al ancho del bus o número de líneas disponibles. La conexión es además bidireccional, lo que permite un ancho de banda de hasta 8GB/seg para un conector 16X, o 16GB/seg para el máximo actual de 32X.

PCI Express incluye características nuevas tales como: manejo de energía, conexión y desconexión en caliente de dispositivos, y la capacidad de manejar transferencias de datos punto a punto, dirigidas desde un host (cualquier nodo en una red que brinda un servicio). Esto último es importante porque permite emular un entorno de red, enviando datos entre dos dispositivos compatibles sin necesidad de que éstos pasen primero a través del host (un ejemplo sería la transferencia directa de datos desde una capturadora de vídeo hasta la tarjeta gráfica, sin que éstos se almacenen temporalmente en la memoria principal). PCI Express también simplifica el diseño de tarjetas madre, pues su tecnología serie requiere un único cable para los datos, frente a los 32 necesarios para el PCI clásico, el cual también necesitaba que las longitudes de estos fuesen extremadamente precisas. La escalabilidad es otra característica clave, pues se pretende que las versiones posteriores de PCI Express sustituyan cualquier característica que PCI o PCI-X pueda ofrecer.

ESTANDARIZACIÓN INTERPLATAFORMA

PCI Express pretende proporcionar consistencia respecto a los conec-

tores y buses entre componentes y plataforma; esto es, el mismo bus y conector se empleará para computadoras de escritorio y portátiles, servidores, y otros dispositivos. Aunque el costo de la implementación de un sistema PCI Express será igual o menor que el de uno PCI, esta estandarización tendrá un precio: aunque será compatible a nivel de software con los protocolos PCI, no lo será a nivel físico con las tarjetas PCI y AGP actuales. Esto supone algunos retos para los fabricantes y distribuidores encargados del cambio, pues el hardware y software debe estar listo en el momento adecuado (si no se realiza una campaña publicitaria masiva mucho antes del lanzamiento, los vendedores podrían encontrarse con un *stock* de tarjetas PCI y AGP a punto de volverse obsoletas y, por tanto, casi imposibles de vender).

PCI Express requiere de soporte en el sistema operativo y Microsoft es el principal fabricante a incorporar sus especificaciones, según el estándar publicadas por Intel y sus asociados, PCI Express debe ser compatible a nivel de software con los actuales sistemas operativos, por lo que la inicialización del sistema, detección del hardware y asignación de recursos no debería suponer un problema. Sin embargo, Microsoft no está publicitando el soporte de PCI Express como una característica principal del Windows de nueva generación (*Longhorn*), y aunque éste y PCI Express estarán hechos el uno para el otro, el sistema operativo no aparecerá hasta 2006, mientras que fabricantes tales como ATI o Nvidia ya están lanzando sistemas PCI Express e intentando que su comercialización sea lo más pronto posible. De hecho, hasta el momento no se ha determinado si los sistemas operativos actuales son totalmente compatibles con las implementaciones de PCI Express.

REVOLUCIÓN GRÁFICA

Sin duda alguna lo más beneficiados por PCI Express serán los fabricantes de tarjetas gráficas. Tales como los ya mencionados fabricantes ATI y Nvidia. El conector PCI Express no es compatible con las tarjetas AGP actuales, por lo que estas tendrán que sustituirse completamente. Por otro lado, también se incluyen mejoras técnicas, tales como una mayor potencia eléctrica disponible. Tal y como demostró la escalada desde AGP 1X hasta 8X, el ancho de banda adicional no ha sido la panacea que se prometía, pero una señal más clara y mayor potencia eléctrica siempre puede suponer una mejora significativa, especialmente en gama alta. Con el bus PCI Express se podrán instalar múltiples tarjetas gráficas de alta gama en el mismo equipo, lo que seguramente abrirá la puerta a innovaciones de software y hardware. El actual AGP 8X (también AGP 3.0) está en los límites del rendimiento; PCI Express resolverá algunos problemas de temporización del AGP actual, y casi triplicará la potencia eléctrica máxima disponible para la tarjeta, lo que lo situará por encima de AGP y del AGP Pro.

PCI Express también supone un impacto en los equipos portátiles. Los estándares PCMCIA y CardBus se sustituirán por ExpressCard (cuyo nombre formal será NewCard), que combina PCI Express y USB 2.0 para permitir una arquitectura modular con mayor ancho de banda. La especificación ExpressCard también admite dispositivos que pueden conectarse mediante el empleo de un adaptador de tamaño completo de 54 mm, o uno fino de 34 mm. El conector físico de PCI Express añade una nueva posibilidad, pues puede tomar forma de un conector clásico o de un cable. La idea del conector-cable ha ido

ganando inercia durante el año pasado gracias a USB 2.0 y FireWire, y su flexibilidad ofrece grandes ventajas.

CONCLUSIONES

Al igual que ocurrió con otros cambios en el sistema de E/S, PCI Express no quitará a sus antecesores de la noche a la mañana. Aún no se puede saber con precisión como será, pero es muy probable que las tarjetas gráficas de alta rendimiento sean las primeras en cambiar, (como es de suponer su naturaleza de alto rendimiento). Suponemos que la sustitución será lenta, en primera instancia, un puente PCI-PCI Express para alojar las tarjetas actuales PCI será una solución inmediata hasta que los fabricantes migren totalmente sus sistemas integrados (Gigabit Ethernet) y externos (almacenamiento extraíble) al nuevo bus.

BIBLIOGRAFÍA

- [1] Mueller, Scott. *"Upgrading and repairing PCs"*, 14th Edition, 2003, Editorial QUE.
- [2] Buchanan, William. *"Computer Busses Design and Application"*, 2000, Editorial CRE.
- [3] Bigelow, Stephen. *"Bigelow's PC Hardware Desk Reference"*, 2003, McGraw Hill.

